DOI: 10.15276/hait.04.2020.4
UDC 004.315

Development of computer system components in critical applications: problems, their origins and solutions

Igor S. Kovalev[1)]
ORCID: 0000-0001-6065-2893, igoryan33@ua.fm
Oleksandr V. Drozd[1)]
ORCID: 0000-0003-2191-6758, drozd@ukr.net
Andrzej Rucinski[2)]
ORCID: 0000-0002-0988-7376, andrzej.rucinski@unh.edu
Myroslav O. Drozd[1)]
ORCID: 0000-0003-0770-6295, myroslav.drozd@opu.ua
Viktor V. Antoniuk[1)]
ORCID: 0000-0001-8436-5338, viktor.v.antoniuk@gmail.com
Yulian Yu. Sulima[3)]
ORCID: 0000-0003-3986-7296, mr_lemur@ukr.net
[1)] Odessa National Polytechnic University, 1, Shevchenko Ave., Odessa, 65044, Ukraine
[2)] University of New Hampshire, Durham, New Hampshire, 03824, Boston, USA
[3)] Odessa Technical Professional College of the Odessa National Academy of Food Technologies, 54, Balkivska St., Odessa, 65006, Ukraine

ABSTRACT

The article is devoted to analysis of problems of the computer system development in the domain of critical applications. The main trends of this development were highlighted, which consisted in increased demands for performance based on parallelization of calculations, processing of approximate data and ensuring functional safety in accordance with the need for structuring for parallelism and fuzziness of the natural world, as well as with increased responsibility in decisions made. Analysis of problems encountered in implementation of existing solutions was carried out. There was a lag behind theories limited by the model of exact data from the practice of processing approximate data for modern systems receiving initial data from sensors, including safety-related systems. The problems of matrix structures, which underlie the design of modern computer systems and demonstrate low efficiency in performance and power consumption, as well as in providing functional safety, important for critical applications, are disclosed. The application of fault-tolerant solutions as the basis of functional safety and distrust of these solutions, which is manifested in the practice of using dangerous imitation modes, were noted. They recreate emergency conditions to improve the checkability in solving the problem of hidden faults, since a fault-tolerant solution does not become fail-safe when there is a shortage of checkability. An analysis was given to the sources of the problems considered and the possibilities of solving them from the point of view of a resource-based approach, which identifies the problem of hidden faults as a challenge of growth with a lag of components from the development of the system. The role of matrix structures in the backlog of components and the need to solve the problem by repeating the version redundancy for these structures are shown. Method of introduction of version redundancy into matrix structure on the basis of strongly connected versions for solution of problems of fault tolerance and checkability in complex is proposed. The effectiveness of the method is estimated on an example of the iterative array multiplier using its software model.

Keywords: computer system; critical application; parallelization of calculations; approximate data; functional safety; fault tolerance; checkability; resource-based approach; problem of hidden faults; matrix structure version redundancy; strongly connected versions

***For citation*: Kovalev I. S., Drozd O. V., Rucinski A., Drozd M. O., Antoniuk V. V., Sulima Y. Y. Development of Computer System Components in Critical Applications: Problems, Their Origins and Solutions. *Herald of Advanced Information Technology*. 2020; Vol.3 No.4: 252–262. DOI: 10.15276/hait.04.2020.4**

INTRODUCTION

The field of critical application of computer systems is constantly expanding along with the quantitative and qualitative growth of high-risk objects, which include power plants and power networks, modern transport infrastructures, various de fense facilities [1]. Under these conditions, computer systems become safety-related systems aimed at ensuring functional safety and a system and a control object to prevent accidents and reduce emergency losses [2].

New directions in the development of information technologies, including Big Data, Cyber-physical systems, the Internet of Things and Everything else, also find application in applications related to ensuring functional security, and therefore become critical.

In these applications, the priority of functional security is combined with high requirements for computing performance and an orientation to the

© Kovalev I. S., Drozd O. V., Rucinski A., Drozd M. O., Antoniuk V. V., Sulima Y. Y. 2020
This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/deed.uk)

processing of approximate data, which is usually performed in floating-point formats [3].

To achieve high performance, arithmetic systems are developed using matrix and pipeline types of parallelism, respectively, at the circuit and system level, that is, systems are designed by pipeline, and sections of the pipeline are single-cycle matrix devices (parallel adders, shifters, iterative array multipliers and dividers). Reception, processing and output of numerical data is performed in parallel codes [4].

The orientation of modern arithmetic systems to the processing of approximate data is dictated by the nature of the initial data. They are the results of measurements and are formed by sensors that make up the lower level in the structure of safety systems and the Internet of Things, as well as Cyber-physical systems.

Functional safety of critical systems is ensured using fault-tolerant solutions, which are aimed at continuing to provide services in case of failures. Arithmetic systems most often use fault-tolerant solutions based on majority structures, which contain several data processing channels and a majority element that selects the result by voting [5].

Special attention in ensuring functional safety is paid to common cause failures, which occur in conditions of copying solutions in channels of fault-tolerant structures [6]. Resistance to common cause failures is based on the use of multi-version technologies and the expansion of the range of types of diversity to minimize common causes [7].

PROBLEMS OF EXISTING DECISIONS

The orientation of modern computer systems to process approximate data and ensure high performance based on system pipeline and circuit matrix parallelism, as well as the use of fault-tolerant solutions in critical applications, has developed historically on the principle of response to emerging needs. The lack of systems in the choice of solutions called into question their effectiveness and required analysis of existing solutions.

In practice, the processing of approximate data has long been a priority in the development of computer systems. However, this development demonstrates a lag in understanding of the processes taking place. In a computer, all data is represented by binary codes, i.e., numbered, element numbers of a set. These numbers, enumerated using ordinal numerals, refer to exact data in nature and constitute the opposite of approximate numbers. The exact number consists only of most significant bits, and the approximate number contains the most and the least significant bits. In these bits, faults cause errors that are respectively essential and inessential for the trustworthiness of the calculated results.

It should be noted that the purpose of on-line testing is to assess the trustworthiness of the results, and not to find a fault, as follows from the definitions of totally self-checking circuits. These goals are the same for exact data, since the detected error indicates both the presence of a fault in the circuit and non-reliable result. For approximate data, targets become significantly different, since the erroneous result is valid in the case of an inessential error. Until now, on-line testing of computer systems retains traditions based on the theory and practice of constructing totally self-checking circuits, which are focused on detecting faults in the calculation process without distinguishing essential and inessential errors [8].

However, inessential errors are much more common than essential ones. Detection of inessential errors results in rejection of valid results. In critical applications, such as radars, the flow of erroneous results can be so large that their recalculation not only reduces performance, but makes it impossible to complete calculations at all.

The use of pipeline parallelism at the system level is fully justified by the simultaneous operation of all sections of the pipeline. Performance losses occur only when the pipeline is started and stopped. When performing tasks without pauses, the pipeline starts and stops once in the entire life cycle [9].

The efficiency of circuit matrix parallelism can be estimated using the example of an iterative array multiplier, given that the multiplication operation plays a special, key role in the processing of approximate data. Indeed, multiplication is used in the floating-point number record itself, and therefore in one form or another is present in all mantissa operations, and the results of these operations inherit the properties of the product. Therefore, the conclusions obtained from the study of multiplication and its execution in matrix form extend to all arithmetic operations and their matrix schemes.

Iterative array multiplier of n-bit binary codes contains matrix of n^2 operational elements and performs operation in one cycle. The duration of the clock cycle is determined by the longest serial connection of operational elements. Such a connection in the fastest scheme contains $2n - 2$ operational elements.

Thus, each of the n^2 operational elements is used in a clock cycle only on its small part $1 / (2n - 2)$, which for $n = 32$ and $n = 64$ is 1.6 % and 0.8 %, respectively [10].

Low utilization of operational elements reduces the ratio of performance to cost. The efficiency of

pipeline parallelism of computer systems is offset by the disadvantages of matrix structures in pipeline sections.

It should also be noted that the remaining clock cycle time is spent on waves of parasitic transitions of signals due to their non-simultaneous propagation along paths of different lengths [11]. It is known that these glitches increase the energy consumed by the 8-bit adder by 30 % [12]. Such an increase in total energy consumption occurs only due to its dynamic component. In the iterative array multiplier, functional transitions to perform the operation occurs at an average of 30 % of the circuit points. Parasitic transitions can occur several times at the points of the circuit, and their number repeatedly exceeds the number of functional transitions.

Thus, for matrix structures, the dynamic component of energy consumption is mainly formed as a result of parasitic signal transitions. The static component is determined by the large dimensions of the matrix circuits.

The disadvantages of matrix structures are not limited to their low efficiency in terms of performance and energy consumption. The main drawback of matrix circuits is manifested in safety-related systems, which differ from ordinary computers by designing for operation in two modes: normal and emergency [2].

Normal mode is the longest. It can occur for decades in anticipation of an emergency mode. At the same time, the emergency mode, as a rule, comes unexpectedly and is little studied. The main experience of the system operation in emergency mode is gained as a result of accidents.

The existence of two modes significantly affects the effectiveness of fault-tolerant solutions. In March 1979, a massive missile attack from the Soviet Union was displayed at three US Air Force command posts. Systems for retaliating were activated. The countdown was suspended according to the results of space reconnaissance, which did not confirm the fact of the attack. Analysis of events showed that at one of the command posts the operator mistakenly inserted a cassette into the system with imitation of the attack. Six months later, the situation repeated: the imitation mode, recreating emergency conditions, turned on due to one burned chip [13].

The planned use of imitation modes is also fraught with increased danger, as it requires temporary shutdown of emergency protection, which was one of the causes of the Chernobyl disaster [14].

The use of imitation modes, the very presence of which poses a real danger to safety-related systems, is explained by the problem of hidden faults. They can accumulate throughout the long-term normal mode and appear with the onset of the most responsible emergency mode, which leads to a decrease in the fault tolerance of the system components.

Thus, the use of imitation modes indicates a distrust of fault-tolerant solutions as a basis for the functional safety of critical systems.

Matrix circuits that process data in parallel codes contribute to the accumulation of hidden faults when limiting a set of input words. In normal mode, circuits often operate with minor changes in input data, for example, at the noise level. Accumulation of faults indicates insufficient checkability of the circuit, i.e. low ability of the matrix circuit to show faults [15].

Simulating the operation of the iterative array multiplier under conditions of limited normal mode input data is shown in Fig. 1.

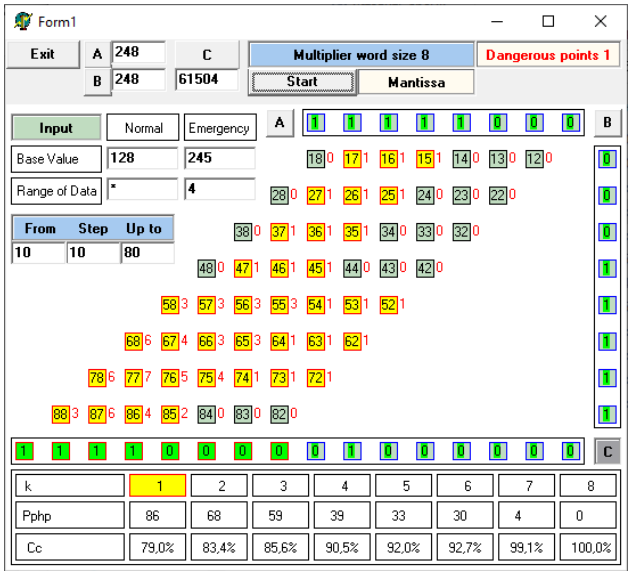


Fig. 1. **Iterative array multiplier operation on restricted normal mode inputs**

Source: **compiled by the author**

A matrix of operational elements, constraint conditions of input data, and experimental results for eight different constraints are represented.

The input data is limited to the mantissa range of both operands. In the first experiment, the mantissas normally take 10 values, forming 100 input words. In each next experiment, the range increases by 10, so that the second experiment uses 20 mantissa values and 400 input words, and in the final eighth experiment – 80 values and 6400 input words.

The purpose of the experiments is to find potentially hazardous points of the multiplier circuit in which the problem of hidden faults can occur, i.e., their accumulation in the normal mode and manifestation with the onset of the emergency mode. Operational elements are colored yellow if there are

ISSN 2663-0176 (Print)
ISSN 2663-7731 (Online)

such points, and to the right of the operational element they are indicated their number.

The table shows the number of potentially hazardous points in the diagram and the evaluation of checkability as an addition to the unit of percentage of these points. When the number of input words decreases from 6400 (39 %) to 100 (0.6 %), the number of potentially hazardous points increases to 86, and the checkability decreases to 79%. When the input data is changed at the noise level, the mantissas change in 1–3 lower bits, which is 4–64 input words.

Problems created by matrix parallelism receive a "simple solution" immediately after realizing their source. This solution is to reduce matrix structures.

Reducing circuits in sections of the pipeline system to one operational element converts this system into a bitwise pipeline for processing data in serial codes. The matrix of such pipelines solves the performance problem in conditions of better use of operational elements in the clock cycle. The causes of parasitic transitions are also eliminated and the dynamic component of power consumption is reduced. And most importantly, serial code leaves no room for hidden faults to accumulate.

However, such a "simple solution" is met with strong opposition from matrix structures, which have dominated for several decades and during this time have significantly strengthened their position, creating a developed supporting infrastructure from models, methods and means, including modern CAD [16].

For example, advanced FPGA design is provided by libraries of ready-made solutions based on matrix structures. The FPGA chip contains a set of built-in iterative array multipliers and preparation for accelerated addition of parallel codes in matrix structures [17].

Thus, the problems of productivity and trustworthiness should be solved within the framework of today's dominance of matrix structures.

The purpose of this paper is to analyze the listed problems with the disclosure of their sources in order to determine possible solutions.

As an example of a possible solution to the problem of low checkability of matrix structures, a method of raising them to the level of diversification is proposed.

SOURCES OF PROBLEMS AND SOLUTIONS

To solve the problems of modern computer systems, it is necessary to study the sources of these challenges. Their analysis can be carried out using a resource-based approach that explores the integration of the computer world into the natural one [18].

Problems are seen as challenges that are provoked in the natural world by ineffective solutions to previous integration problems. The solution to the problem is provided by three components: the performance you need to achieve for the full amount of work, the trustworthiness of the result, and the investment of resources to achieve performance and trustworthiness. Resources include everything necessary to solve the problem, that is, everything laid down into three components: models, methods and means (materials and tools).

Models are our ideas about the natural world and its details, including the artificial world created by man. Methods serve to develop and evaluate resources. Models and methods form the information part of resources. Means are their material carriers, representing models and methods in their structure and functioning. The structure of the woke marmot describes the model of the coming spring – a protracted cold or aimed at a hot summer. In addition, the sleeping marmot demonstrates the method of saving energy, widely adopted in green technologies.

The human mission is to read information resources from the material carriers of the natural world and write them open source to the material carriers of the computer world. The recording process is preceded by understanding the resources read, developing and verifying them by creating means which are also an incentive for the development of models and methods that are most appropriate to the natural world.

The resource-based approach identifies three levels of resource development: replication, diversification and self-sufficiency as the goal of development.

Replication marks the lowest level of resource development that will always be chosen if resource niches are opened: market, technological, environmental and others. Open niches mean no conflict with the natural world, no restrictions on the replication of resources. The success of integration at this level is associated with productivity gains. In the natural world, excess fertility over mortality due to productivity is the main way to survive for most of the simplest biological forms.

With the closure of resource niches, replicated clones can survive only by showing features, that is, becoming individuals, versions, rising to the level of diversification, where integration into the natural world is due to increased trustworthiness, which is adequacy in relation to this world.

At this level, integration occurs in contact with the natural world and consists in structuring

resources according to its characteristics, among which the computer world has shown the most parallelism and fuzziness.

The most striking example of structuring is the development of personal computers, which in hardware support for approximate computing have gone from the optional supply of Intel287/387 coprocessors to several floating-point pipeline units (FPU – Floating-Point Units) in the Pentium family central processor and several thousand such pipelines in the graphics processor and CUDA technology their use for parallel computing [19].

The problem of approximate data follows from the initial accurate and consistent understandings (models) and the capabilities (methods) of a human that he demonstrates in resource development. A human is praised for his accuracy and consistency. However, the objective course of this development is aimed at integration into a parallel and fuzzy natural world.

The exact data were structured into floating-point formats, which represent a number in the form of two components: mantissa and exponents [3]. This diversification allowed to spread the requirements for the accuracy and range of the values of the number. In the exact number, these characteristics are rigidly connected. In the natural world, quantitative estimates require large ranges with little accuracy. The size of the mantissa and exponents independently determine the accuracy and range of the values of the number.

Diversification identified in the approximate number (mantissa) the Most and Least Significant Bits (MSB and LSB) [20]. Errors caused by circuit failures are diversified in these bits into essential and inessential ones for the trustworthiness of the result. The need for interaction between the mantissa and the exponent gave rise to operations of renormalization of operands and normalization of the result [3].

The problem of matrix structures is their belonging to the lower level of development – replication. At this level, you can increase performance by scaling matrix structures, but you cannot improve reliability, including safety. In the absence of formative contact with the natural world, there is no basis on which adequacy to this world can be stimulated.

Calculations can be replicated for the organization of a double account or majority structures. But verification of the result or the selection of the right one from a set of replicated results can be carried out only at the level of diversification due to version redundancy, when the results are calculated in several versions. In an approximate world, nothing is replicated completely the same, versions are replicated. Next, the question is versioning redundancy indicators – whether it is enough for the required level of adequacy. The development of understanding of problems to the level of diversification is encouraged in this world by simplifying the solution and improving its quality. Our replication layer models interfere this process, for example, when numbering depersonalizes entities and thus deprives them of their features.

The problem of matrix structures in safety-related systems should be solved by the development of version redundancy [21].

The problem of fault-tolerant solutions in critical applications (the problem of hidden faults) is the impossibility of converting such solutions to fail-safe solutions in case of lack of checkability.

Checkability is best known as the testability of digital circuits, which, in fact, is structural checkability, since it is completely determined by the structure of the circuit [22]. In the operating mode, the checkability also depends on the input data and becomes structurally functional.

Computer systems are transformed into safety-related ones by diversifying the operating mode into normal and emergency. Following the mode, the input data and the checkability of the circuit are diversified, which become different in normal and emergency modes.

Thus, the problem of hidden faults is a challenge of growth when a computer system in critical applications reaches a level of diversification in checkability, and the components of this system continue to duplicate at the level of replication with the dominance of matrix structures. There is no problem with hidden faults on normal computers because they remain hidden throughout the operating mode, which is the only one.

Understanding problems provides the key to solving them:

• elevate components to the system level in the problem of hidden faults as a challenge of growth;

• raise matrix structures to the level of diversification, developing version redundancy of fault-tolerant solutions;

• develop pipeline systems for critical use in floating-point formats in accordance with the requirements of parallelism and approximation of the natural world.

DEVELOPMENT OF THE VERSION REDUNDANCY IN MATRIX STRUCTURES

The known use of version redundancy relates to majority structures, where the natural form of this redundancy present in duplicated channels serves to

ISSN 2663-0176 (Print)
ISSN 2663-7731 (Online)

develop fail-safe systems that resist independent failures. In critical systems, version redundancy develops with the introduction of various types of diversity to counter the common cause failures that occur when copying matrix circuits. The main limitation in the development of such multi-version technologies is complexity, which is growing along with the number of versions implemented on resource-consuming matrix structures [23].

Hidden faults, which pose a real functional safety risk no less than common cause failures, can also be eliminated using version redundancy, which raises matrix structures to the level of diversification.

A method of creating versions in a matrix structure based on Strongly Connected Versions (SCVs) is proposed, providing minimal additional redundancy [24].

The method uses the particularities of matrix structures that perform arithmetic operations. Such structures repeat the features of positional number systems and therefore demonstrate the uniformity of elements and the regularity of connections. The multi-bit parallel adder consists of a set of equally series-connected full adders, and the matrix multiplier comprises a set of equally series-connected columns or rows of the matrix of operational elements. The regularity of connections is broken only at the boundaries of arithmetic schemes. In SCV structures, regularity is restored by closing the matrix circuit, supplemented by one unified element, into a ring. In addition, the ring is supplemented by elements of its break between each pair of adjacent unified elements.

Fig. 2 shows the SCV structure.

The circuit comprises a calculation unit 1 in the form of a matrix converted into a ring, registers of operands 2 and a result 3, which receive *n*-bit operands at the input *O* and send an *n*-bit result to the output *R*. In addition, the circuit uses the monitoring, status (counter) and control (decoder) units 4, 5 and 6, respectively.

For a matrix structure consisting of *n* uniform OE elements, the ring contains $v = n + 1$ such elements and *v* break elements D. These elements are controlled by a decoder 6 which generates a unitary code containing a single unit value to break the ring at a given location. The decoder 6 is controlled using a counter 5 by modulo *v*, which in each clock cycle forms *v* values defining a sequence of ruptured ring states before the next uniform element OE. The version of the matrix of *n* unified elements and the additional element at the *v* position is formed starting from the element following it. Registers 2 and 3 establish a correspondence between the input/output data and the matrix version using cyclic shifters to form operands and select the result.

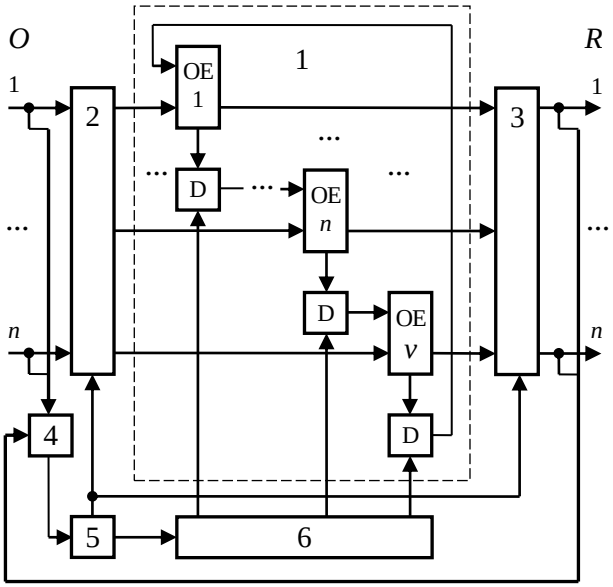


Fig. 2. **SCV Structure Diagram**
Source: **compiled by the author**

Cyclic shifters add an additional position *v* to the operand and eliminate it from the result. The OE elements corresponding to this position are excluded from the calculation.

The counter 5 in each cycle goes into the next state and thus provides a change in the versions on which the operation is performed.

Operands and result are supplied to unit 4, which checks their compliance. In the event of a detected error, counter 5 continues to change the state until the correct version is available to prevent the failure from affecting the operation. Further calculations are performed on this correct version, and in the case of a new error caused by the same fault, on the next correct version.

The constant version change allows to regularly shift the information content of the matrix structure relative to the structure itself and thus raise it to the level of diversification necessary to ensure the checkability of the circuits with low diversity of the input data presented in parallel codes.

EVALUATION OF THE METHOD

The proposed method can be evaluated at a qualitative level in comparison with known and promising solutions. A quantitative estimate can be obtained by examining the specific application of the method.

The main qualitative assessment should be given to the method from the point of view to receive a solution important for ensuring functional safety in critical applications. Traditional multi-

version systems are aimed at obtaining a fault-tolerant solution by increasing the level of diversity in the versions used. Such a solution is implemented by increasing the structural redundancy of matrix circuits and, accordingly, reducing their checkability, which is necessary for converting a fault-tolerant solution to fail-safe one. The proposed method does not oppose fault tolerance to checkability, but provides these functional safety requirements in the complex.

The checkability of the circuits is achieved by a continuous version change requiring additional power consumption compared to fixed matrix structures, since the dynamic component of energy consumption is proportional to the number of signal transitions. This problem is significant for autonomous systems and can be solved by diversifying transitions by distinguishing between zero-to-one and one-to-zero transitions. Currently, they are not distinguished, and the dynamic component of energy consumption is proportional to their sum, and in the future, it should become a difference close to zero, since the transition from one-to-zero should return energy in the same way as the operation of an electric car.

The quantification of the method can be obtained using an iterative array multiplier.

The SCV structure can work with or without stopping the input stream when an error is detected. In these cases, system recovery occurs with a decrease in performance or trustworthiness of the results, respectively. A new error may occur when the input data changes, which may indicate or hide the fault.

The main indicator of a solution with reduced performance is the number of clock cycles spent on restoring the system. The decrease in trustworthiness can be estimated by the number of unreliable results calculated during the fault elimination process.

These indicators were evaluated by developing a software model that reproduces the functioning of the matrix structure of the multiplier with continuous version change until the correct version is obtained after the fault is detected.

In each cycle, the software model randomly determines the input data of the multiplier, as well as the location (operation element) and the short type of fault. The operation element circuit contains a full adder and a gate AND. A fault of the short type connects two points of the circuit, which are the inputs and outputs of the full adder and the gate AND. The fault can be masked or manifested in the form of an error that is essential or in essential for the result, and can also be detected or missed by monitoring. The program in each clock cycle simulates the calculation of the product on a new version. At the beginning of the next cycle, the calculation space is shifted from the matrix of elements by one row and one column.

During the experiment, the program calculates and averages the number of clock cycles spent on restoring the iterative array multiplier and the number Z of the calculated reliable results, i.e., results that do not contain essential errors. The results are monitored by modulo three, which detects all errors caused in the iterative array multiplier by the considered faults. This conclusion was obtained during the experiment. The experiment was conducted for *n*-bit iterative array multipliers, where $n = 8, 9, ..., 15$.

The results of the experiments are shown in Table 1, where T is the average number of clock cycles for restoring the iterative array multiplier when the input data is stopped; $R_1 = Z_n / T\ 100\%$ and $R_2 = Z_{n-2} / T\ 100\%$ – trustworthiness of results in the process of restoration in cases of determination of Z_n and Z_{n-2} values of Z number for n and $n - 2$ MSB.

Table 1. **Simulation Results**

n	8	9	10	11	12	13	14	15
T	2.4	2.5	2.7	2.9	3.3	3.8	4.3	4.7
R_1, %	52.3	53.0	53.2	53.2	53.3	53.3	53.4	53.4
R_2, %	74.4	72.6	70.8	69.1	67.6	66.5	65.9	65.3

Source: **compiled by the author**

As n grows, the average number of T clock cycles increases from 2.4 to 4.7. The trustworthiness of the results R_1 slightly increases, and R_2 decreases from 74.4 % to 65.3 %.

CONCLUSION

The main challenges in the development of modern computer systems are their structuring for parallelism and fuzziness of the natural world, as well as increasing the responsibility of decisions made in accordance with the rapid development of critical applications. Still, the development trend continues to have high performance requirements based on parallelization of calculations and the already fairly declared dominance of approximate data in calculations. This is evidenced by the development of the Internet of Things and Everything, Cyber-physical systems and safety-related systems that receive initial data from sensors. The requirements for functional safety, traditionally provided on the basis of the use of fault-tolerant solutions, come to the fore.

The practice of using simulation modes shows distrust of fault-tolerant solutions that do not become fail-safe in conditions of insufficient checkability and create a problem of hidden faults.

The resource-based approach identifies this problem as a challenge of growth, when in critical applications the system rises to the level of diversification, but its components still continue to be stamped using matrix structures, related to a lower level of replication.

ISSN 2663-0176 (Print)
ISSN 2663-7731 (Online)

The dominance of matrix structures in models, methods and means, including CAD, limits the ability to solve the growth challenge. Today, the level of component development needs to be improved within matrix structures by improving version redundancy.

Traditional fault-tolerant solutions based on multi-version technologies oppose fault tolerance to checkability of circuits.

The proposed method of introducing version redundancy into matrix structures solves the problems of fault tolerance and checkability in the complex.

An experimental test of the method using the example of a iterative array multiplier showed the restoration of the system in case of failure in an average of 2.4 – 4.7 clock cycles with the calculation of more than half of the reliable results (from 52.3 % to 74.4 %) during the elimination of the fault.

REFERENCES

1. Ivanchenko, O., Kharchenko, V., Moroz, B., Kabak, L. & Konovalenko, S. “Risk Assessment of Critical Energy Infrastructure Considering Physical and Cyber Assets: Methodology and Models”. *In Proc. of the 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications.* Lviv. Ukraine: 2018. 225–228. DOI: 10.1109/IDAACS-SWS.2018.8525594.

2. Smith, D. & Simpson, K. “The Safety Critical Systems Handbook”. [5th ed.]. Butterworth-Heinemann: 2019. DOI: https://doi.org/10.1016/C2019-0-00966-1.

3. 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. URL: http://www.dsc.ufcg.edu.br/~cnum/modulos/Modulo2/IEEE754_2008.pdf.

4. Palagin, A. & Opanasenko, V. “The implementation of extended arithmetic’s on FPGA-based structures”. *In: Proceedings of the International Conference IDAACS.* Bucharest. Romania: 2017; p.1014–1019. DOI: 10.1109/IDAACS.2017.8095239.

5. Volochiy, B., Yakubenko, V. & Zmysnyi, M. “The Reliability Model of Fault-Tolerant System with the Majority Structure and Considering the Change in the Failure Rate of the Core Module During Operation”. *15th IEEE International Conference TCSET.* Lviv-Slavsko. Ukraine: 2020. DOI: 10.1109/TCSET49122.2020.235532.

6. IEC 62340:2007. Nuclear power plants – Instrumentation and control systems important to safety – Requirements for coping with common cause failure. Geneva: IEC, 2007. URL: https://standards.iteh.ai/catalog/standards/iec/60e552e0-a682-4fa1-86b9-ce37edc38809/iec-62340-2007.

7. Kharchenko, V., Bakhmach, E., Siora, A., Sklyar, V. & Tokarev, V. “Diversity-Oriented FPGA-Based NPP I&C Systems: Safety Assessment, Development, Implementation”. *18th International Conference on Nuclear Engineering.* Xi’an. China: 2010; Vol.1: 755–764. DOI: 10.1115/ICONE18-29754.

8. Metra, C., Schiano, L., Favalli, M. & Ricco, B. “Self-checking scheme for the on-line testing of power supply noise”. *Design, Automation and Test in Europe Conference*. Paris. France: 2002. p. 832–836. DOI: 10.1109/DATE.2002.998395.

9. Ramamoorthy, C. V. & Li, H. F. “Pipeline Architecture”. *Computing Surveys*. 1977; Vol.9 No.1: 61–102.

10. Drozd, J., Drozd, A., Antoshchuk, S., Kushnerov, A. & Nikul, V. “Effectiveness of Matrix and Pipeline FPGA-Based Arithmetic Components of Safety-Related Systems”. *The 8th IEEE International Conference IDAACS.* Warsaw. Poland: 2015. p.785–789. DOI: 10.1109/IDAACS.2015.7341410.

11. Warren, S. & Anderson, J. “FPGA Glitch Power Analysis and Reduction”. *In: International Symposium on Low power electronics and design.* Fukuoka. Japan: 2011. p.27–32. DOI: 10.1109/ISLPED.2011.5993599.

12. Chandracasan, A. P., Sheng, R. & Brodersen, S. “Low-Power CMOS Digital Design”. *IEEE Journal of solid-state circuits.* 1992; Vol. 27. 4. 473–484. DOI: 10.1109/4.126534.

13. Gillis, D. “The Apocalypses that Might Have Been”. [Electronic resource]. – Access mode : https://www.damninteresting.com/the-apocalypses-that-might-have-been/. 2007. – (Active link: 2019.03.20).

14. Blakemore, E. “The Chernobyl disaster: What happened and the long-term impacts”. 2019. URL: https://www.nationalgeographic.com/culture/topics/reference/chernobyl-disaster.

15. Drozd, O., Kharchenko, V., Rucinski, A. et. al. “Development of Models in Resilient Computing”. *In: 10th IEEE International Conference on Dependable Systems, Services and Technologies.* Leeds, UK. 2019. p.2–7. DOI: 10.1109/DESSERT.2019.8770035.

16. “Xilinx ISE Design Suite”. [Electronic resource]. – Access mode: https://www.xilinx.com/products/design-tools/ise-design-suite.html. 2019. – (Active link: 2019.03.20).

17. Amano, H. “Principles and Structures of FPGAs”. *Publ. Springer.* Singapore: 2018. DOI: 10.1007/978-981-13-0824-6.

18. Drozd, J. & Drozd, A. “Models, Methods and Means as Resources for Solving Challenges in Co-Design and Testing of Computer Systems and their Components”. *9th International Conference on Digital Technologies 2013.* Zhilina. Slovak Republic: 2013. p.225–230. DOI: 10.1109/DT.2013.6566307.

19. “NVIDIA CUDA Compute Unified Device Architecture”. [Programming Guide / Version 1.0, NVIDIA Corporation]. 2007. URL: https://fdocuments.in/document/nvidia-cuda-compute-unified-device-architecture-a-set-of-simd-multiprocessors-with.html.

20. Akinola, S. & Olatidoye, A. “On the image quality and encoding times of LSB, MSB and combined LSB-MSB steganography algorithms using digital images”. *International Journal of Computer Science & Information Technology.* 2015; Vol.7 No. 4:79–91. DOI: 10.5121/ijcsit.2015.7407.

21. Drozd, O., Romankevich, V., Kuznietsov, M. et. al. “Using natural version redundancy of FPGA projects in area of critical applications”. *In: Proceedings of the 11th IEEE International Conference DESSERT.* Kyiv. Ukraine: 2020. p.58–63. DOI: 10.1109/DESSERT50317.2020.9125050.

22. Matrosova, A., Nikolaeva, E., Kudin, D. & Singh, V. “PDF testability of the circuits derived by special covering ROBDDs with gates”. *IEEE East-West Design and Test Symposium, EWDTS* Rostov-on-Don. Russian Federation: 2013.p.1–5. DOI: 10.1109/EWDTS.2013.6673183.

23. 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.* 2002; Vol. 63 No.6: 991–1003. DOI: https://doi.org/10.1023/A:1016130108770.

24. Walkowiak, T., Mazurkiewicz, J., Sugier, J., Zamojski, W. (Eds.). “Monographs of System Dependability”. Dependability of Networks. Wroclaw. Poland: 2010. URL: https://publications.hse.ru/en/books/51732992.

Conflicts of Interest: the authors declare no conflict of interest

Received 05.10.2020
Received after revision 09.11.2020
Accepted 20.11.2020

DOI: 10.15276/hait.04.2020.4
УДК 004.315

Розвиток компонентів комп’ютерних систем в критичних застосуваннях: проблеми, їх джерела та рішення

Ігор Станіславович Ковальов[1)]
ORCID: 0000-0001-6065-2893, igoryan33@ua.fm
Олександр Валентинович Дрозд[1)]
ORCID: 0000-0003-2191-6758, drozd@ukr.net
Анджей Русінський[2)]
ORCID: 0000-0002-0988-7376, andrzej.rucinski@unh.edu
Мирослав Олександрович Дрозд[1)]
ORCID: 0000-0003-0770-6295, myroslav.drozd@opu.ua
Віктор Вікторович Антонюк[1)]
ORCID: 0000-0001-8436-5338, viktor.v.antoniuk@gmail.com
Юліан Юрійович Суліма[3)]
ORCID: 0000-0003-3986-7296, mr_lemur@ukr.net
[1)] Одеський національний політехнічний університет, 1, проспект Шевченка, Одеса, 65044, Україна
[2)] Університет Нью-Гэмпшира, Дарем, Нью-Гэмпшир, 03824, Бостон, США
[3)] Одеський технічний фаховий коледж Одеської національної академії харчових технологій, 54, Балківська вулиця, Одеса, 65006, Україна

АНОТАЦІЯ

Стаття присвячена аналізу проблем розвитку комп'ютерних систем в домені критичних додатків. Виділено основні тренди цього розвитку, які полягають в підвищених запитах до продуктивності на основі розпаралелювання обчислень, до обробки наближених даних і забезпечення функціональної безпеки відповідно до необхідності структурування під паралелізм і наближеність природного світу, а також до підвищення відповідальності щодо прийнятих рішень. Проведено аналіз проблем, що виникають при реалізації існуючих рішень. Відзначено відставання теорій, обмежених моделлю точних даних, від практики в обробці наближених даних для сучасних систем, які отримують вихідні дані від датчиків, включаючи

ISSN 2663-0176 (Print)
ISSN 2663-7731 (Online)

системи критичного застосування. Розкрито проблеми матричних структур, які лежать в основі проектування сучасних комп'ютерних систем і демонструють низьку ефективність у продуктивності та енергоспоживанні, а також у забезпеченні функціональної безпеки, важливої для критичних додатків. Відзначено застосування відмовостійких рішень як основи функціональної безпеки і недовіру до цих рішень, яке проявляється в практиці використання небезпечних імітаційних режимів. Вони відтворюють аварійні умови для підвищення контролепридатності у вирішенні проблеми прихованих несправностей, оскільки відмовостійке рішення не стає відмовобезпечним при дефіциті контролепридатності. Дан аналіз джерел розглянутих проблем і можливостей їх вирішення з позиції ресурсного підходу, який ідентифікує проблему прихованих несправностей як проблему зростання з відставанням компонентів від розвитку системи. Показана роль матричних структур у відставанні компонентів і необхідність вирішення проблеми шляхом розвитку версійної надмірності для цих структур. Запропоновано метод введення версійної надмірності в матричну структуру на основі сильно пов'язаних версій для вирішення проблем відмовостійкості та контролепридатності в комплексі. Ефективність методу оцінена на прикладі матричного помножувача з використанням його програмної моделі.

Ключові слова: комп’ютерна система; критичне застосування; розпаралелювання обчислень; наближені дані; функціональна безпека; відмовостійкість; контролепридатність; ресурсний підхід; проблема прихованих несправностей; матрична структура; версійна надмірність; сильно пов'язані версії

DOI: 10.15276/hait.04.2020.4
УДК 004.315

Развитие компонентов компьютерных систем в критических приложениях: проблемы, их истоки и решения

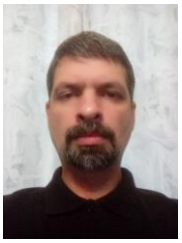
Игорь Станиславович Ковалев[1)]
ORCID: 0000-0001-6065-2893, igoryan33@ua.fm
Александр Валентинович Дрозд[1)]
ORCID: 0000-0003-2191-6758, drozd@ukr.net
Анджей Русинский[2)]
ORCID: 0000-0002-0988-7376, andrzej.rucinski@unh.edu
Мирослав Александрович Дрозд[1)]
ORCID: 0000-0003-0770-6295, myroslav.drozd@opu.ua
Виктор Викторович Антонюк[1)]
ORCID: 0000-0001-8436-5338, viktor.v.antoniuk@gmail.com
Юлиан Юрьевич Сулима[3)]
ORCID: 0000-0003-3986-7296, mr_lemur@ukr.net
[1)]Одесский национальный политехнический университет, 1, проспект Шевченко, Одесса, 65044, Украина
[2)]Университет Нью-Гэмпшира, Дарем, Нью-Гэмпшир 03824, Бостон, США
[2)]Одесский технический профессиональный колледж Одесской национальной академии пищевых технологий, 54, Балковская улица, Одесса, 65006, Украина

АННОТАЦИЯ

Статья посвящена анализу проблем развития компьютерных систем в домене критических приложений. Выделены основные тренды этого развития, заключающиеся в повышенных запросах к производительности на основе распараллеливания вычислений, к обработке приближенных данных и обеспечению функциональной безопасности в соответствии с необходимостью структурирования под параллелизм и приближенность естественного мира, а также с повышением ответственности в принимаемых решениях. Проведен анализ проблем, возникающих при реализации существующих решений. Отмечено отставание теорий, ограниченных моделью точных данных, от практики в обработке приближенных данных для современных систем, получающих исходные данные от датчиков, включая системы критического применения. Раскрыты проблемы матричных структур, которые лежат в основе проектирования современных компьютерных систем и демонстрируют низкую эффективность в производительности и энергопотреблении, а также в обеспечении функциональной безопасности, важной для критических приложений. Отмечено применение отказоустойчивых решений как основы функциональной безопасности и недоверие к этим решениям, которое проявляется в практике использования опасных имитационных режимов. Они воссоздают аварийные условия для повышения контролепригодности в решении проблемы скрытых неисправностей, поскольку отказоустойчивое решение не становятся отказобезопасным при дефиците контролепригодности. Дан анализ источникам рассмотренных проблем и возможностей их решения с позиции ресурсного подхода, который идентифицирует проблему скрытых неисправностей как проблему роста с отставанием компонентов от развития системы. Показана роль матричных структур в отставании компонентов и необходимость решения проблемы путем развития версионной избыточности для этих структур. Предложен метод введения версионной избыточности в матричную структуру на основе сильно связанных версий для решения проблем отказоустойчивости и контролепригодности в комплексе. Эффективность метода оценена на примере матричного умножителя с использованием его программной модели.

Ключевые слова: компьютерная система; критическое применение; распараллеливание вычислений; приближенные данные; функциональная безопасность; отказоустойчивость; контролепригодность; ресурсный подход; проблема скрытых неисправностей; матричная структура; версионная избыточность; сильно связанных версии

ABOUT THE AUTHORS



Igor S. Kovalev – Master, Senior Lecturer of Computer Intellectual Systems and Networks Department, Odessa National Polytechnic University. Odessa, Ukraine
***Research field*:** On-Line Testing; Green Technologies and Circuit Checkability in the Digital Component of Safety-Rlated Systems; LUT-oriented Architecture of FPGA-Based Systems

Ігор Станіславович Ковальов – старший викладач кафедри Комп'ютерних інтелектуальних систем та мереж. Одеський національний політехнічний університет. Одеса, Україна

Игорь Станиславович Ковалев – старший преподаватель кафедры Компьютерных интеллектуальных систем и сетей. Одесский национальный политехнический университет. Одесса, Украина

Oleksandr V. Drozd – Dr. Sci. (Eng.), (2003), Prof. of Computer Intellectual Systems and Networks Department. Odesa National Polytechnic University. Odesa, Ukraine.
***Research field*:** On-Line Testing; Green Technologies and Circuit Checkability in the Digital Component of Safety-Rlated Systems; LUT-oriented Architecture of FPGA-Based Systems

Олександр Валентинович Дрозд – доктор технічних наук (2003), професор кафедри Комп'ютерних інтелектуальних систем та мереж, Одеський національний політехнічний університет. Одеса, Україна.

Александр Валентинович Дрозд – доктор технических наук (2003) профессор кафедры Компьютерных интеллектуальных систем и сетей, Одесский национальный политехнический университет. Одесса, Украина

Andrzej Rucinski – PhD Professor Emeritus, Department of Electrical and Computer Engineering. University of New Hampshire, a member of the Executive Committee (Innovation Chair) of the IEEE Computer Society's Design Automation Technical Committee, USA Ambassador of International Society of Service Innovation Professionals. Boston, USA
***Research field*:** Ecosystem Grand Challenges Associated with eHealth/mHealth; eEducation/eLearning; eSecurity/Identity Protection; Smart City/Region/State and Information Infrastructure Technologies Involving a Digital Ecosystem Using Internet of Things

Анджей Русінський – почесний доктор філософії, кафедра Електротехніки та обчислювальної техніки. Університет Нью-Гэмпшира, член Виконавчого комітету (голова з інновацій) Технічного комітету з автоматизації проєктування IEEE Комп'ютерного товариства, посол Міжнародного товариства професіоналів у області сервіс-інновацій. Бостон, США

Анджей Русинский – почетный доктор философии, кафедра Электротехники и вычислительной техники. Университет Нью-Гэмпшира, член Исполнительного комитета (председатель по инновациям) Технического комитета по автоматизации проектирования IEEE Компьютерного сообщества, посол Международного общества профессионалов в области сервис-инноваций. Бостон, США

Myroslav O. Drozd – PhD (2014), Associated Prof. of Information Systems Department. Odesa National Polytechnic University. Odesa, Ukraine
***Research field*:** On-Line Testing and Circuit Checkability in the Digital Component of Safety-Related Systems

Мирослав Олександрович Дрозд – кандидат технічних наук (2014), доцент кафедри Інформаційних систем. Одеський національний політехнічний університет. Одеса, Україна

Мирослав Александрович Дрозд – кандидат технических наук (2014), доцент кафедры Информационных Систем. Одесский национальный политехнический университет. Одесса, Украина

Viktor V. Antoniuk – PhD (2020), Associated Prof. of Computer Intellectual Systems and Networks Department. Odesa National Polytechnic University. Odesa, Ukraine
***Research field*:** On-Line Testing of the Digital Components; FPGA-Based Systems

Віктор Вікторович Антонюк – кандидат технічних наук (2020), доцент кафедри Комп'ютерних інтелектуальних систем та мереж. Одеський національний політехнічний університет. Одеса, Україна.

Виктор Викторович Антонюк – кандидат технических наук (2020), доцент кафедры Компьютерных интеллектуальных систем и сетей. Одесский национальный политехнический университет. Одесса, Украина

Yulian Yu. Sulima – PhD (2014), Head of the Computer Systems Department, SSU "Odessa Technical Professional College of the Odessa National Academy of Food Technologies". Odessa, Ukraine
***Research field*:** Technology of Designing Computer Systems on FPGA; Computer Systems for Critical Application; Checkability and Detection of Hidden Faults of Integrated Circuits

Юліан Юрійович Суліма – кандидат технічних наук (2014), завідувач відділення комп'ютерних систем ВСП «Одеський технічний фаховий коледж Одеської національної академії харчових технологій». Одеса, Україна

Юлиан Юрьевич Сулима – кандидат технических наук (2014), заведующий отделением компьютерных систем ОСП «Одесский технический профессиональный колледж Одесской национальной академии пищевых технологий». Одесса, Украина

ISSN 2663-0176 (Print)
ISSN 2663-7731 (Online)