Increasing Survivability of Technological Systems Based on the Technology of Programmable Logic Device

Serhii Shtanenko[a], Yurii Samokhvalov[b], Serhii Toliupa[b] and Oleksiy Silko[a]

[a] *Heroes of Kruty Military Institute of Telecommunications and Information Technologies, Moskovska str. 45/1, Kyiv, 01011, Ukraine*
[b] *Taras Shevchenko National University of Kyiv, Volodymyrs'ka str. 64/13, Kyiv, 01601, Ukraine*

Abstract

The article presents an approach of constructing highly reliable (survivable) technological systems, based on the technology of programmable logic device, PLD. The following approach gives the possibility to implement the multi-processor data processing together with the possibility of paralleling calculations, multiple use of elements of reconfigurable integrated circuits, which PLD is, as well as their remote reprogramming. And this, in its turn, makes it possible to increase the survivability and, as a consequence, the reliability of not only computing (microprocessor) systems, programmable logic controllers and instrumentations, but the entire technological system as a whole.

Keywords [1]

Technological system, computing (microprocessor) system, programmable logic integrated circuit, reliability, survivability.

1. Introduction

The intensive development and improvement of digital devices lead to their active use as control systems in various fields of science and technology: transport, energy, economy, telecommunications, military etc. It is especially worth noting the technological process control systems used in the aerospace industry, aviation, military equipment, nuclear power plant management, medicine and other industries. The disruption of such crucial control systems can cause significant economic or ecological damage, threaten human health or life and sometimes even lead to catastrophic consequences [1].

The technological process control system (hereinafter-technological system) will be understood as an automated or automatic system, which is a set of equipment, facilities, complexes and systems of data processing, transmission and reception. This system is constructed for organizational management and/or control of technological processes (including industrial, electronic, communication equipment, other technical and technological means) regardless of the system access to the Internet and/ or other global networks [2].

It should be noted that the basis for constructing technological systems (TS) today is computer (microprocessor) systems (CS), programmable logic controllers (PLC), control and measuring instruments, which belong to the class of complex technical systems (CTS), the behavior of which is difficult to model because of complex dependencies between their parts or because of complex interactions between this system and the environment [3-5].

The importance of the tasks solved by modern TSs, on the one hand, and the complexity of such systems, on the other hand, requires from such systems high reliability and survivability.

Nowadays, the key instrument to the theory of reliability is the reserved copying and, in contrast to it, the survivability of the TS is estimated by its redundancy. Survivability, in its turn, allows to widen

Information Technology and Implementation (IT&I-2021), December 01-03, 2021, Kyiv, Ukraine
EMAIL: sh_sergei@ukr.net (A. 1); yu1953@ukr.net (B. 1); tolupa@i.ua (B. 2); silko.av@gmail.com (A. 2)
ORCID: 0000-0001-9776-4653 (A. 1); 0000-0001-5123-1288 (B. 1); 0000-0002-1919-9174 (B. 2); 0000-0003-4275-4305 (A. 2)

the theory of reliability determining the system stability at the level of its structure, both at the stages of construction and modeling, as well as at the stage of operation [6]. The improving survivability is usually carried out by special mechanisms of adaptation, reconstruction, reconfiguration and reorganization [7]. This makes it possible to create structures that ensure the performance of a critical subset of functions to achieve the purpose of functioning.

Today, a great number of studies are devoted to the issues of TS survivability. Thus, the paper [8] presents some deterministic and stochastic models of restorable and unrestorable systems, which give the possibility to discover the changes in survivability properties with time.

The paper [9] presents the methods of assessing functional and structural survivability based on the use of game-theoretical and entropic approaches, as well as logical and probabilistic models. It considers [10] the problem of increasing structural survivability of info-communication network by optimal structural redundancy carried out on the basis of prior estimation of significance or priority of flows using the corresponding path sets.

It should be noted that in the considered works the estimation of structural survivability of TSs is mainly carried out by classical methods and probability theory models. This does not take into account the factor of adverse effects associated, for example, with cyber attacks on the system.

Recently, to improve the reliability and survivability of TSs, the approach based on the use of programmable logic devices (PLD) as a modern element base, rather than universal processors with "rigid" structure, microcontrollers and custom large integrated circuits is widely used. These systems refer to the reconfigurable devices, capable of changing their integral logical structure directly in the process of operation [11, 12]. Thus, the use of PLD makes it possible to create (design) computing (microprocessor) systems which architecture is focused on solving the particular application problem or class of problems [13, 14]. The article also presents an approach to the creation (design) of highly reliable (survivable) computing (microprocessor) systems, programmable logic controllers and instrumentation in the basis of PLD, which form the basis of constructing modern technological systems operating under the conditions of adverse effects, including cyber attacks.

2. The problem of increasing the survivability of technological systems.

As it was stated above, the main task of TS is to control various processes, including technological process, as well as to improve the efficiency of their managing by minimizing human involvement into these processes.

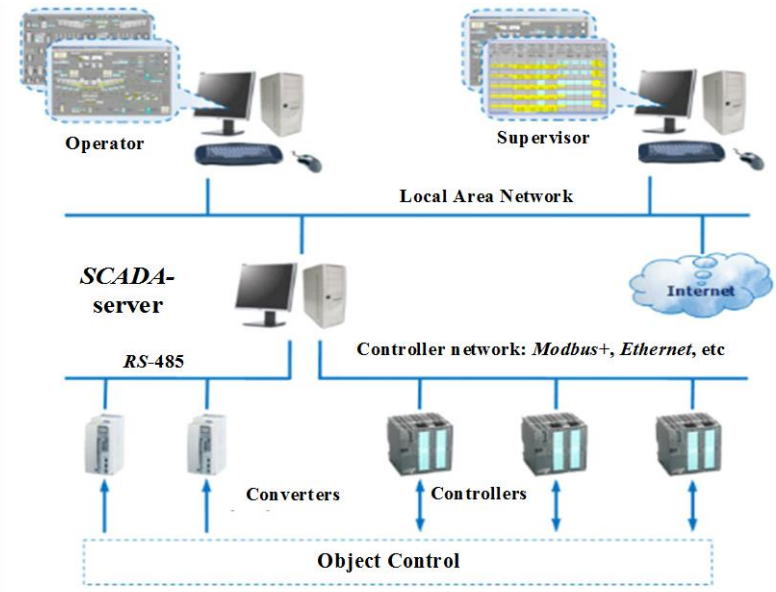

Figure 1: Technological system structure

It should be noted that the hierarchical principle of construction is implemented in modern TSs. According to it, the TS at the upper level of management can be built from the separate subsystems of

the TS, which in its turn, are functionally subdivided into two levels: object-oriented (lower) and instrumental (upper) levels (Fig.1).

Object-oriented subsystems of the lower level are designed to solve the following tasks: real-time measurement of object parameters; object control; collection of measurement data; efficient data processing, temporary storage, presentation of data to the operator and sending them to the TS instrumental level. In this case programmable logic controllers, instrumentation, automation devices, actuators, alarm panels are used as the element base.

Instrumental top-level subsystems of the upper level, which can be attributed to SCADA systems (Supervisory Control And Data Acquisition) are designed to solve the following problems: preparation and debugging of programs and sending them to the object subsystems; object subsystems management; information processing in time sharing mode; accumulation and long-term storage of large amounts of information; documentation of research results. In this case, the basis of the upper level makes computing (microprocessor) systems with a 'rigid' internal structure [15].

A computing (microprocessor) system is understood a set of interconnected and interacting processors or computers, peripheral equipment and software, designed to collect, store, process and distribute information [16].

It should be noted that the constant complication of computing (microprocessor) structures and conditions of their operation, as well as the responsibility of the functions performed by modern systems, due to the development of TS, bring the problems of reliability and survivability to the forefront.

Reliability theory operates with a number of indicators such as: probability of no-failure operation, average time between failures, availability factor, average recovery time, failure rate, etc. These indicators describe the process of functioning well enough only when possible failures and breakdowns of structural elements can be somehow foreseen and described in the form of some probable distributions. If the occurring faults and failures of structural elements cannot be foreseen, for example, due to adverse effects, including cyber attacks, the mechanisms of reliability theory in this case are ineffective [17, 18].

Due to the above said, new characteristics (like functional, effective reliability, software reliability) are constantly introduced into the reliability theory in order to solve the problems of describing the functioning of complex structures, such TS.

At the same time these characteristics describe simple structures well and provide a basis for a complete analysis of its functioning. Based on this analysis they can effectively use well-known approaches to increase reliability, namely:

improvement of characteristics of constituent elements, creation of fundamentally new elements with higher reliability;

using different methods of organizing their structures (control and self-control of computational structure elements, introduction of structural redundancy, use of intermediate and final results correction schemes).

Currently, among the methods of improving reliability, the main place is occupied by backup coping. However, such methods as reconfiguration of computational (microprocessor) systems, reorganization of solution algorithms, multiple use of structure elements are not yet applied in the classical reliability theory.

The disadvantages of the current state of reliability theory, its characteristics and methods include the fact that it operates with failures, faults, failures in complex systems, which are described by some laws of probability distribution.

Proceeding from the above, for a more complete description of the state of modern computing systems as the basis for building modern control systems and expanding the area of described failures, a qualitatively new approach is necessary, in particular, to assess the impact of failures on the functioning of structures and methods to combat them. One of such approaches can be the use of evaluation of computational structures in terms of their survivability, i.e. the ability to withstand any failures of the constituent elements without any limitations of their distribution in time [7].

Survivability is the property of a complex system to adapt under changing conditions of functioning, to withstand adverse influences and to achieve the goal of functioning by changing its behavior and structure. Adverse influences are considered to be possible failures, failures and malfunctions of hardware and software, catastrophic impacts of natural or man-made origin, and it is

not the nature of the impact, but its consequences, as well as cyber attacks on the system are important [9].

Thus, nowadays, the provision and improvement of survivability (functional, structural and informational) of the TS are proposed to implement by means of the developed mechanisms of recognition, counteraction, recovery, as well as by special means of adaptation, reconstruction, reconfiguration and reorganization [19]. Currently, the basis for the organization of recognition mechanisms are diagnostic methods. Mechanisms of counteraction are based on the backup coping methods. Recovery mechanisms are developed on the basis of a range of specialized hardware and software methods and recovery tools.

And if the mechanisms of recognition, counteraction and recovery at a sufficient level are defined and implemented, the special means (adaptation of reorganization, reconfiguration and reconstruction) today require special description and development at the stage of TS design.

Thus, according to [20, 21], special means to improve the survivability of TS are proposed to be implemented at the level of the element base of technical devices. Therefore, further improvement of survivability of modern computing (processing) systems, as well as programmable logic controllers and instrumentation, which form the basis of TS, are associated with the construction of such an element base, which is able to implement the mechanisms of adaptation, reconstruction, reconfiguration and reorganization. This will improve the structural survivability of the TS in the event of adverse effects, including cyber attacks, due to the implementation of multiprocessor data processing, paralleling of calculations and remote reprogramming of the computing (microprocessor) system.

2.1. Approaches to the construction of the element base of technical means.

Currently, the element base of electronic equipment for information processing and storage are integrated circuits (ICs), which are divided into digital, analog and analog-digital. In its turn, digital ICs are divided into standard and specialized (Fig.2)

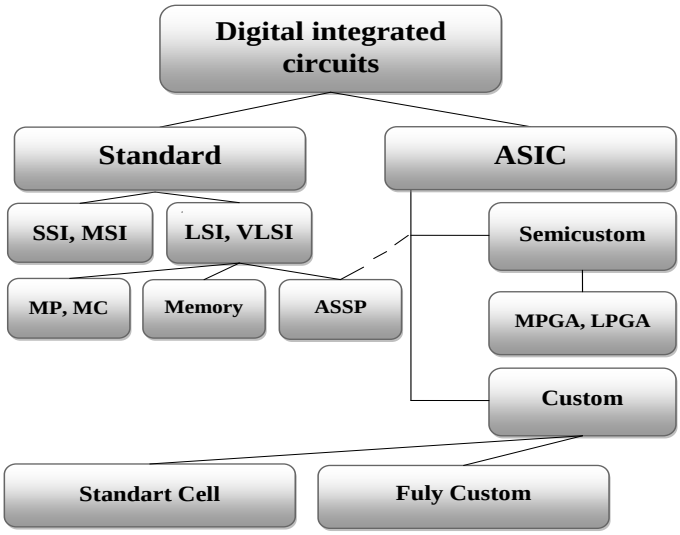

Figure 2: Digital integrated circuit classification

Standard ICs have almost a rigid internal structure, without affecting the nature of their functioning. Specialized ICs (ASIC – Application-Specific Integrated Circuit), have individual nature of functioning and are developed (designed) according to a specific order. The most important achievement in this direction has been the emergence of ICs with a programmable structure.

Integrated circuits with programmable structure (programmable logic devices – PLD) is a matrix of programmable logic elements with SPLD (Simple Programmable Logic Devices), CPLD (Complex Programmable Logic Device), FPGA (Field Programmable Gate Array), FLEX (Flexible Logic Element Matrix) structures.

Unlike conventional digital microchips, the operating logic of PLD is not defined during manufacturing, but is set by programming (design). For programming it they use an integrated development environment (IDE – Integrated Development Environment) that allows to specify the desired structure of a digital device in the form of a circuit diagram or a program in special hardware description languages AHDL, VHDL, Verilog.

The advantages of PLD include: [22].

universality, i.e. possibility to create practically any digital device in a crystal at presence of a personal computer and corresponding tools (CAD);

the possibility of modifying projects at any stage of development and during its operation;

high speed, low power of consumption and high reliability provided by the technology of crystal fabrication;

environmental compatibility through a choice of supply voltage levels and I/O signal parameters;

low project implementation cost as compared to large integrated circuits due to mass production of crystals with regular structure and small time, spent on project development and their verification.

On the basis of the considered structures (SPLD, CPLD, FPGA and FLEX) not only combinational and consistent digital devices, digital machines MEALY and MOORE, but also computing (microprocessor) systems are designed by implementing one of the levels of design: low, block or high.

Low level involves the use of hardware description languages AHDL, VHDL, Verilog, which manage the development of a digital device at the register-transfer level (RTL). In this case, registers are formed, (similar to the processor), and logical functions, that change the data between them, are defined.

At the block level there is a connection of library program-like IP-blocks (Intellectual Property), which carry out the certain functions for reception of the necessary functionality of system on a chip (System-on-Chip, SoC) (Fig.3).

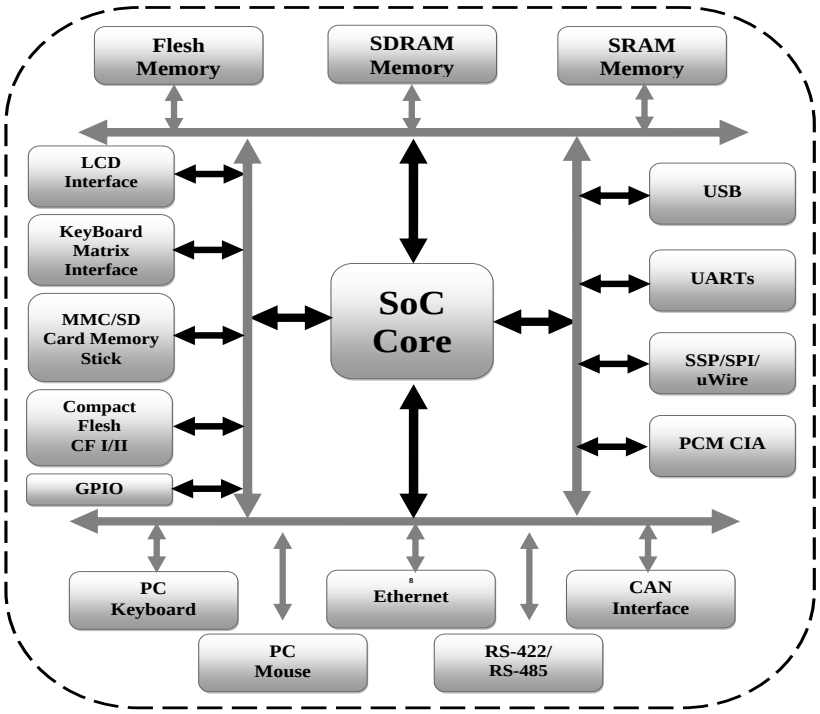

Figure 3: The structure of a typical SoC

System-on-Chip or SoC is a computing (microprocessor) system, the structure of which is designed specifically to solve an applied problem (or a group of problems). System-on-Chip technology or SoC is implemented as a set of functionally specialized hardware and software components based on a configurable microelectronic platform [23]. Moreover, this technology represents two independent functional parts PLD FPGA (Field-Programmable Gate Array) and HPS (Hard Processor System), interconnected by data exchange interface.

At high level of design high-level programming languages, like C/C++, System C, Python, Java, MATLAB are used, which at the level of abstraction, i.e. introduction of semantic structures, briefly describe structures of computing (microprocessor) systems and operations over them. And with the help of compilers and transmitters (HLS for C/C++, MyHDL for Python etc) give the opportunity to transmit the written blocks (structures) to the Verilog/VHDL hardware description languages to the RTL-level of register transfers.

The rapid development of PLD technology has led to the fact that they are now successfully competing with general-purpose microprocessors, microcontrollers and signal processors in the fields of control and high-speed data processing, digital signal processing, cryptography and other fields [24].

Reconfiguration is a significant advantage of PLD in systems where continuous availability is a critical factor. If one PLD component fails, reconfiguration can be performed to restore the performance by replacing the failed component with a copied one. Therefore, many leading electronics manufacturers use PLD as coprocessors to universal microprocessors or as additional modules in a multiprocessor system to solve time-consuming tasks for a universal processor [25].

One of the promising directions of improving the survivability and reliability of modern computing (microprocessor) systems is the creation of PLD-based multiprocessor system with structure DIMIMD (Distributive Implementation Multiple Instruction stream, Multiple Data stream) [26] This structure is an improved structure MIMD (Multiple Instruction stream, Multiple Data stream) and is a continuation of the technology of recursive computing systems with non-traditional (different from 'rigid' von-Neumann or Harvard) structure.

Computing (microprocessor) system with DIMIMD structure belongs to the class of multiprocessor systems with a large number of processors (single-type modules), each of which can independently execute the program and which under the control of operating system can be combined for joint solution of one task. Besides, this system, including several copies of each resource (crucial and specialized processors, operating and external memory, input-output devices etc.) is able to provide high survivability and reliability parameters and consequently, reliability with the right choice of structure as well as the organization of technical and mathematical facilities. The basis for ensuring the survivability and reliability parameters is gradual degradation of redundant system characteristics as its elements fail.

The ideology of gradual degradation is based on the availability of redundant copies of each type of resource, on the ability of the system to detect the failed element and exclude it from its composition. In addition, the ability to change the set of elements allocated to solve the problems (reduce their number or change the physical binding), i.e. to reconfigure the system.

Given the fact that the computing (microprocessor) system with the structure DIMIMD refers to self-healing systems, an additional requirement is the ability to diagnose a failed element, carrying out some work in restoring it, regardless of the solutions of the problem and their subsequent inclusion in the number of active elements. Thus, an effective means of monitoring and maintaining the performance of a multiprocessor system can be a service processor (readiness management device) that provides the collection of information about the state of the system's processors and the automatical reconfiguration of the system, eliminating the faulty processor or device.

One of the main features of PLD is the ability to use the principle of parallel data processing (computation paralleling) to solve a wide range of problems [27]. Increasing the resources of modern programmable logic and reducing their cost allowed, due to the transition from quantity to quality, to increase the speed of developed digital devices significantly and to implement hardware algorithms that operate in real time.

Paralleling of calculations or logical operations can be performed both at the level of information representation bits and at the level of blocks, performing the required algorithms of a mathematical model.

Thus, the prototype for organizing parallel computing in multimodular systems, in which modules were elementary machines with data storage, processing and transportation capabilities, was multimodular multi-transcomputing systems (transputer), which gave a subsequent impetus to the development of System-on-Chip technology, SoC [28]. Multi-transcomputing systems have on-chip memory and built-in efficient interfacing facilities. This allows different structures corresponding to subsets of valid configurations to be combined within a single model (a single chip). Besides, a

certain commonality of the offered methods and means of organizing parallel data processing is connected with the representing of parallel algorithms in the form reflecting first of all "inner" parallelism of the task and not the structure of the system on which the corresponding programs will be performed.

The high degree of inner parallelism of tasks allows us, when managing parallel computations, to match not only the technical system's structure to the parallel algorithm's system but also the algorithm to the system's structure. The mentioned properties, on the one hand, provide a real opportunity to build reconfigurable computing systems and parallel algorithms and, on the other hand, provide a methodological basis for a rather universal approach when creating specialized forms of data processing organization based on private configurations of systems and algorithms [29].

Another important feature of PLD is adaptation (reconfiguration) to the process being executed, i.e. the ability to change the algorithms of operation depending on change conditions or requirements. This implies that the algorithm(s) in the form of a file loaded into the chip should be made in advance. The loaded file is the result of the design and verification process of the device that hardware implements a given algorithms [30].

On the basis of multiprocessor technology and reconfigurable, programmable, integrated circuits, which PLD are, dynamically reconfigurable systems (DRS) are built. These computing systems are capable of changing their internal logical structure directly in the process of operation for a time that is much shorter than the execution time of computational tasks, between which the change of structure took place [31]. At the same time, the maximum allowable configuration time T_{CONF}^{TH} for DRS in the general case can change $T_{CONF}^{TH} = f(t)$. Taking this into account, the system is dynamically reconfigurable if the following inequalities are valid:

$$\begin{cases} T_{CONF\,i}^{TH} << T_{TSK\,i}, \text{ where } i = 1, \\ T_{CONF\,i}^{TH} << \min\left\{T_{TSK\,i-1}, T_{TSK\,i}\right\}, \text{ where } i = \overline{2, n} \end{cases} \quad (1)$$

where $T_{TSK\,i}$ is the execution time i-th computational task.

Therefore, whether the system is dynamically reconfigurable depends as much on the characteristics of the hardware platform as on the tasks to be solved. A typical DRS consists, as a rule, of two basic elements (Fig.4):

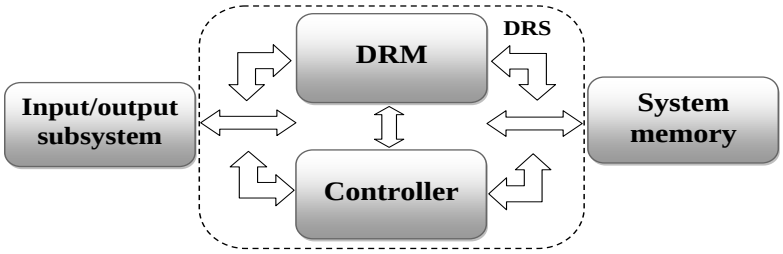

Figure 4: General block diagram DRS

dynamic reconfigurable module (DRM) – calculator, which is designed to implement computational operations;

control module (controller), in which the basic control operations and a special operation of loading the configuration file to the dynamically reconfigurable module from an arbitrary place in memory are implemented. In this case the controller is actually a processor, the functional basis of which may be incomplete (all computational operations, except those, required when working with memory, may be absent).

The dynamically reconfigurable module is the element that distinguishes DRS from traditional processor devices, although the realization of such a fundamental property of computing devices as dynamic reconfiguration is only possible with both elements.

Thus, dynamically reconfigurable information processing systems are a set of structures focused on the implementation of technical devices with the possibility of adaptation and, as a consequence, make it possible to create on their basis highly reliable (survivable) computing (microprocessor) systems. In this case the classical processors with "rigid" internal structure are only a special case of

DRS that has significant limitations in functionality. Further development of PLD-based DRS technology allows the transition to the design of adaptive computing devices capable of changing their internal structure depending on the implemented functions and tasks [31].

3. Conclusion

The paper considers the problem of improving survivability of technological systems under adverse effects on the system. An analysis of existing methods for improving the reliability and survivability of complex systems is performed, advantages and disadvantage of the methods are noted. A variant of technological design of computing (microprocessor) systems based on PLD technology with specialized reconfigurable integrated circuits as a basis is proposed. In general, the considered option allows, through the implementation of special means of adaptation, reconstruction, reconfiguration and reorganization of the structures, to create a high reliability (survivability) technological system that can operate in conditions of adverse effects, and in conditions of cyber attacks.

4. References

[1] L. Slipachuk, S. Toliupa and V. Nakonechnyi, "The Process of the Critical Infrastructure Cyber Security Management using the Integrated System of the National Cyber Security Sector Management in Ukraine", 2019 3rd International Conference on Advanced Information and Communications Technologies (AICT), 2019.

[2] Zakon Ukrayiny № 2163 VIII ot 05.10.2017 «Pro osnovni zasady zabezpechennya kiberbezpeky Ukrayiny».

[3] Bani Younis, M. & Frey, G. (2006). UML-Based Approach for the Reengineering of PLC Programs, in Proceedings of 32nd Annual Conference of the IEEE Industrial Electronics Society (IECON'06), pp. 3691–3696, Paris, France, November, 2006.

[4] R. S. Moura, F. C. Alves do Couto and L. A. Guedes, "Control and plant modeling for manufacturing systems using statecharts," 2008 IEEE International Symposium on Industrial Electronics, 2008, pp. 1831–1836, doi: 10.1109/ISIE.2008.4676971.

[5] Naumenko, E.M., Samokhvalov, Yu.Ya. Formation of a rational variant of hardware realization of automated control systems at the stages of pre-project design. Upravlyayushchie Sistemy i Mashiny, 2005, (5), pp. 62 – 67.

[6] V. R. Westmark, "A definition for information system survivability," 37-th Annual Hawaii International Conference on System Sciences, 2004. Proceedings of the, 2004, pp. 10 pp., doi: 10.1109/HICSS.2004.1265710.

[7] Gulyayev V.A. Organizatsiya zhivuchikh vychislitel'nykh struktur / V. A. Gulyayev, A. G. Dodonov, S. P. Pelekhov. – Kiyev: Nauk. dumka, 1982. – 138 s.: il.; 20 sm.

[8] S. Dick, C. L. Bethel and A. Kandel, "Software-Reliability Modeling: The Case for Deterministic Behavior," in IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans, vol. 37, no. 1, pp. 106 – 119, Jan. 2007, doi: 10.1109/TSMCA.2006.886364.

[9] Dodonov A.G. Vvedeniye v teoriyu zhivuchesti vychislitel'nykh sistem/A.G. Dodonov, M.G. Kuznetsova, Ye.S. Gorbachik. – K.: Nauk. dumka, 1990. – 184 s.

[10] Shi Jianxu and J.P. Fonseka, "Traffic-based survivability analysis of telecommunications networks", Global Telecommunications Conference, vol. 2, pp. 79–87, 1995.

[11] Robert J. Ellison and Andrew P. Moore, "Architectural Refinement for the Design of Survivable Systems", SEI, Oct 2001.

[12] Y. Jin, N. Satish, K. Ravindran, and K. Keutzer, "An automated exploration framework for FPGA-based soft multiprocessor systems," in Proceedings of the 3rd IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and Systems Synthesis (CODES+ISSS '05), pp. 273–278, ACM, Jersey City, NJ, USA, September 2005.

[13] R. Joost and R. Salomon, "Advantages of FPGA-based multiprocessor systems in industrial applications," 31st Annual Conference of IEEE Industrial Electronics Society, 2005. IECON 2005., 2005, pp. 6 pp. –, doi: 10.1109/IECON.2005.1568946.

[14] Qin Xia, Zhiwen Wang, Restoring Information System Survivability Based on Component Availability, Cyber Technology in Automation Control and Intelligent Systems (CYBER) 2011 IEEE International Conference on, pp. 220–225, 2011.
[15] Carlos Queiroz, Abdun Mahmoud, Zahir Tari, SCADA Survivability: An Analytical Framework Using Performance Modeling, Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE, pp. 1–5, 2010.
[16] Michael J. Spinks. Microprocessor System Design: A Practical Introduction. 2013. p. 158.
[17] K. Kyamakya, K. Jobman and M. Meincke, "Security and survivability of distributed systems: an overview", 21st Century Military Communications Conference Proceedings, vol. 1, pp. 1204 – 1208, 2000.
[18] R.J. Ellison, D.A. Fisher, R.C. Linger, H.F. Lipson, T.A. Longstaff and N.R. Mead, "Survivability: protecting your critical systems", IEEE Internet Computing, vol. 3, no. 6, pp. 9.B.3-1-9.B.3-8, Nov-Dec 1999.
[19] Li Bai, Saroj Biswas, Albert Ortiz, Frank Ferrese, Don Dalessandro, Qing Dong, "Survivability Analysis of Reconfigurable Systems," IEEE International Conference 2007, Industrial Engineering and Engineering Management, pp. 663–667, 2007.
[20] Zhiwen Wang, Ke Lu, Huafeng Zou, Qin Xia, Resilience Recovery Based on Component Availability in Information Systems, Fifth Frontier International Conference of Computer Science and Technology (FCST), 2010, pp. 221–226, 2010.
[21] S. Jha and J.M. Wing, "Survivability analysis of networked systems", Proceedings of the 23rd International Conference on Software Engineering, pp. 872–874, 2001.
[22] S. Dick, C. L. Bethel and A. Kandel, "Software-Reliability Modeling: The Case for Deterministic Behavior," in IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans, vol. 37, no. 1, pp. 106–119, Jan. 2007, doi: 10.1109/TSMCA.2006.886364.
[23] W. Wolf, A. A. Jerraya, and G. Martin, "Multiprocessor System-on-Chip (MPSoC) technology," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 27, no. 10, pp. 1701–1713, 2008.
[24] C.-Y. Tseng and Y.-C. Chen, "Design and implementation of multiprocessor System on a Chip (mpSoC) based on FPGA," in Proceedings of the International Computer Symposium (ICS '09), 2009.
[25] Kalyayev I.A., Levin I.I., Semernikov Ye.A., Shmoylov V.I. Rekonfiguriruyemyye mul'tikonveyyernyye vychislitel'nyye struktury / I.A. Kalyayev, I.I. Levin, Ye.A. Semernikov, V.I. Shmoylov. – Rostov-na-Donu: Izd-vo YUNTS RAN, 2008. – 320 s.
[26] Pogrebinskiy S. B. Proyektirovaniye i nadezhnost' mnogoprotsessornykh EVM / S. B. Pogrebinskiy, V. P. Strel'nikov. – M.: Radio i svyaz', 1988. – 168 s.
[27] M. Hubner, K. Paulsson, and J. Becker, "Parallel and flexible multi-processor System-on-Chip for adaptive automotive applications based on Xilinx microblaze Soft-cores," in Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium, p. 149a, April 2005.
[28] D. Göhringer, B. Liu, M Hübner, and J. Becker, "Star-wheels Network-on-Chip featuring a self-adaptive mixed topology and a synergy of a circuit – and a packet-switching communication protocol," in Proceedings of the 19th International Conference on Field Programmable Logic and Applications (FPL '09), pp. 320–325, September 2009.
[29] D. Göhringer, T. Perschke, M. Hübner, and J. Becker, "A taxonomy of reconfigurable single-/multiprocessor Systems-on-Chip," International Journal of Reconfigurable Computing, vol. 2009, Article ID 395018, 11 pages, 2009.
[30] C. Claus, W. Stechele, and A. Herkersdorf, "Autovision: a run-time reconfigurable mpSoC architecture for future driver assistance systems," Information Technology Journal, vol. 49, no. 3, pp. 181–187, 2007.
[31] Filippov A.K. Adaptive Computing Systems: Definition and Mathematical Background / A.K. Filippov // Proceedings of the 8th International Conference "Perspective technology in the mass media – PTMM'2009". – 2009. – P. 86 – 89.