Original article
DOI: 10.21683/1729-2646-2018-18-3-10-17

Simulation model of dependability of redundant computer systems with recurrent information recovery

Igor V. Egorov, *St.Petersburg State Polytechnic University, Saint Petersburg, Russia*

Igor V. Egorov

Abstract.*Today's digital nanotechnology-based information management systems are especially sensitive to highly-energized particles during operation in irradiated areas. This sensitivity is most often manifested in the form of intermittent soft errors, i.e. distortion of information bits in the system's memory elements with no hardware failure. The cause is in the afterpulses at the output of the logical elements that occur as the result of ionization of the gate area of the transistor's semiconductor after it is exposed to a highly-energized particle. In order to counter soft errors the system is equipped with self-repair mechanisms that ensure regular replacement of distorted data with correct data. If this approach to design is employed, the significance of dependability analysis of the system under development increases significantly. Since regular occurrence of soft errors is essentially normal operating mode of a system in conditions of increased radiation, dependability analysis must be repeatedly conducted at the design stage, as that is the only way to duly evaluate the quality of the taken design decisions. The distinctive feature of fault-tolerant hardware and software systems that consists in the presence of nonprobabilistic recovery process limits the applicability of the known methods of dependability analysis. It is difficult to formalize the behaviour of such systems in the form of a dependability model in the context of the classic dependability theory that is geared towards the evaluation of hardware structure. As it has been found out, the application of conventional methods of dependability analysis (such as the Markovian model or probabilistic logic) requires making a number of assumptions that result in unacceptable errors in the evaluation results or its inapplicability.* **Aim.** *Development of the model and methods of dependability analysis that would allow evaluating the dependability of hardware and software systems with periodic recovery.* **Results.** *A simulation model was developed that is intended for dependability evaluation of complex recoverable information management systems. The model is a network of oriented state graphs that allows describing the behaviour of a recoverable system subject to the presence of computation processes and recovery processes that operate according to non-stochastic algorithms. Based on the simulation model, a software tool for dependability analysis was developed that enables probabilistic estimation of dependability characteristics of individual system units and its overall structure by means of computer simulation of failures and recoveries. This tool can be used for comprehensive dependability evaluation of hardware and software systems that involves the analysis of recoverable units with complex behaviour using the developed simulation model, and their operation along with simple hardware components, such as power supplies and fuses, using conventional analytical methods of dependability analysis. Such approach to dependability evaluation is implemented in the Digitek Reliability Analyzer dependability analysis software environment.* **Practical significance.***The application of the developed simulation model and dependability analysis tool at the design stage enables due evaluation of the quality of the produced fault tolerant recoverable system in terms of dependability and choose the best architectural solution, which has a high practical significance.*

Keywords: *dependability model, simulation model, soft error, dependability theory, recoverable system.*

For citation: *Egorov I.V. Simulation model of dependability of redundant computer systems with recurrent information recovery. Dependability 2018;3: 10-17. DOI: 10.21683/1729-2646-2018-18-3-10-17*

Introduction

Out of [1–3] dedicated to the analysis of effects in semiconductor structures exposed to radiation follows that a highly-energized particle hitting a MOS transistor can cause the ionization of the gate area of the semiconductor. Due to that, the output of a gate that includes a transistor may produce as short false signal (voltage pulse), of which the duration is usually within 1 to 2 ns. In the context of today's nanotechnology-based integrated circuits such induced false pulses present a danger, since their characteristics are comparable with those of the useful signals and can cause distortions of information in the computer system.

If the false pulse changes the state of a trigger or another storage element, the event usually called soft error occurs. It consists in the fact that from the dependability point of view it can cause a failure, since a change in the internal state of system's memory affects its operation. At the same time, the equipment in this case remains operational, which means that the state of the system can be recovered by overwriting distorted data with correct ones.

Studies in the area of improvement of radiation durability of information management systems [4-6], including those conducted by a group of researchers of the St.Petersburg State Polytechnic University [7-11], show the introduction of periodic recovery facilities is the solution that enables a qualitative improvement of a system's resistance to soft errors.

The operation of such self-repairing systems has a number of distinctive features that affect the method of evaluation of their dependability and make the conventional methods of dependability analysis hardly applicable [12-16]. Dependability analysis for such systems is of utmost importance in the design process, as occasional soft errors are essentially part of their normal operation. Consequently, the design of this type of systems is impossible without detailed estimation of dependability that allows evaluating the quality of the structure under development. For this reason the development of new models and methods of dependability analysis that would cover the distinctive features of the self-repaired systems resistant to soft errors is now a relevant task.

Conventional analytical models of dependability of computer systems and their limitations

Over the years of dependability theory development many models and methods of dependability analysis were constructed. Most of them are geared towards the solution of the following practical problem: provided that a certain hardware system operates in stationary mode, when, with time, its individual components randomly fail, to estimate the system's time to failure and to identify the most structurally important components in terms of dependability.

However, when conducting dependability analysis, systems with periodic recoverythat operate in conditions of regular soft errors, the following features must be taken into consideration:

• the mechanism of memory state recovery implies that the distorted information bits are periodically rewritten. Obviously, recovery does not occur randomly, but at determined moments in time;

• the analyzed systems are hardware and software systems, in which the software component (computational process) often defines their operation. The hardware component, in turn, can be considered as a resource that must be in a operable state at the moment the computational process refers to it. Besides the primary computational process, there is a recovery process that at specific moments also requests access to the resource (memory).

Both of the above features complicate the requirements for the design of highly dependable systems. On the one hand, the designer must ensure the shortest possible period of recovery in each of the system's units. On the other hand, the recovery process must not block the resources required by the primary computational process. These contradictory requirements must be taken into consideration both during system design (synthesis) and it dependability analysis: having information on the operating algorithm of the primary computational process, the maximum permissible period of recovery in the units can be defined and dependability of the designed system can be estimated subject to the specified conditions. If the dependability requirements are not observed, as early as at the design stage the system architecture must be modified in favour of additional dependability improving solutions [7]. Under this approach the dependability analysis is the tool of a fault-tolerant system synthesis.

In practice, the following conventional analytical methods of dependability estimation are used:

1) combinatory estimation.

For a recoverable hardware unit with a fixed recovery period, the possible combinations of events (component failures) and the effects of such events on other connected components are analyzed. As the result, a probability function is constructed that connects the failure rates of the unit's components with the failure rate of the unit itself. For generic structures the formula is known in advance and it is sufficiently simple to substitute into it the parameters failure occurrence and moments of recovery [13]. This estimation procedure dictates a limited number of considered events occurring over the recovery period, in order to considerably reduce the number of analyzed combinations and thus simplify the final expression, which causes a growing error in the results.

2) Markovian model

If the combinations of component failures that occurred in the system are identified as system states, and all the failure and recovery events are associated with transitions between such states, the system can be represented

as a Markovian model. In this case the result is obtained by solving a system of Chapman-Kolmogorov algebraic equations[16]. The moments of all transitions must follow the exponential law of random distribution, which causes error that can be quite considerable [12]. Also, as the number of system components grows, the number of states in the model increases greatly.

3) logical and probabilistic method.

The logical and probability function of the system operability is constructed by well-known methods [17,18]. Its construction is also bound by limitations on the distribution law of failure and recovery moments typical to the application of the Markovian model.

Since all the above methods have limitations in terms of the analyzed systems, cause errors in estimation results, as well as are quite tedious in practice,it appears to be advisable to develop software tools that would enable automated dependability evaluation of systems with periodic recovery after soft errors.

Simulation model of dependability of a recoverable computer system

Since the analyzed systems have fairly complicated behaviour, it appears that simulation models, rather than analytical evaluation methods, are more applicable in their dependability assessment. At the same time, the use of general-purpose simulation models (such as the Petri nets) does not appear to be practically applicable, as it requires significant labour contributions from the user (system designer) in order to construct an adequate model. A specialized simulation model that would operate such dependability theory terms as "failure", "recovery", but would allow simulating a wide range of structures, seems to be appropriate. The software tools that operate this model must automatically calculate the desired dependability characteristics, such as the operability function and mean time to failure. This approach will enable quick modifications of the simulation model and recalculation of dependability characteristics subject to their changes, which is especially important at the design stage.

For this purpose the author has developed a dependability simulation model that is based on the representation of the system as an oriented state graph that contains the following basic elements:

• states that are defined by the set of failed components. Each state is classified as operable or inoperable (in this case states of "soft", i.e. recoverable error and unrecoverable error should be distinguished);

• transitions between states that usually occur in case of soft errors or unrecoverable failures or recoveries after soft errors.

Transitions between states may occur either at random or determined moments in time. Therefore, for each transition, a distribution law of the random value of occurrence (normal distribution, exponential distribution or determined moment of occurrence) and distribution characteristics (event rate) are defined. In the process of simulation, the events associated with the state (that occur a certain time upon transition into such state) and those not associated with a specific event must be distinguished. For example, the moment of recovery is not associated with the current state, as it occurs with a fixed rate independent of the moment of failure of any element that caused the system's transition into the current state. For the simulation of such events, a special entity called the Global Events Generator was introduced at model level. It contains the description of the rules occurrence of all events that do not depend on the current state of the system.

The analyzed system may contain a significant number of elements, and each state of the simulation model in general is based on the sum of the states of all of its elements. This causes a significant growth of the number of states and complication of the model. In order to solve this problem, the model may be described not as a single state graph, but in the form of a **network** consisting of multiple graphs. Transitions in each graph of this network may occur:

• due to an event associated with the current graph state;

• upon reception of signal from the Global Events Generator;

• upon occurrence of an event in another graph of the network (such events are called external).

In order to define the condition, under which the system is considered inoperable, in the model, parameter Health Function must be defined. The health function in the context of the simulation model is represented in the form of enumeration of graphs that must be in operable state for the system to be deemed operable.

The software tool developed by the author that operates this simulation model works as follows. The description of the simulation model (network of graphs) loads from xml files, after which the number of experiments specified in the models' parameters is performed. In the course of each experiment, the occurrence of random and determined events described in the model is simulated, and the time to system failure (moment, when at least one of the graphs enumerated in the operability function is inoperable) is measured. Since the experiment simulates random events, the estimate is also a random value. In order to evaluate the estimation error, the specified number of experiments (model parameter) is simulated, based on which the standard deviation of the estimate's random value is calculated. Provided the scope of statistics collected out of experiment results is sufficient, the probability function of system operability of time can be evaluated.

An example of application of the simulation model for dependability evaluation of a recoverable unit of a computer system

As an example, let us evaluate the time to failure of a unit of an information management system that operates with the clock cycle T and has the following structure:

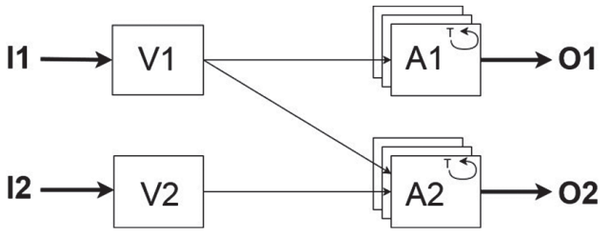

Figure 1. Analyzed unit of information management system

A1, A2 are functional modules, each of which is triplexed and majorized in order to improve the dependability, as well as have inbuilt mechanisms that ensure recovery of the module's state at each cycle (with the frequency 1/T). The modules ensure setting of output signals O1 and O2 respectively.

V1, V2 are voting components that ensure setting of correct input data I1 and I2 for components A1 and A2 per 2-out-of-3 voting rule.

O1, O2 are output signals of the unit, of which the correctness determines the operability of the whole unit. Since A1 and A2 are triplexed and majorized, the data of the corresponding outputs O1 and O2 become incorrect if 2 and more instances of modules A1 and A2 are inoperable. As long as only one instance is exposed to soft error, the module is in the degraded state, but this does not affect the system's operability in general.

The unit is affected by a flow of soft errors, as the result of which the triplexed instances of modules A1 and A2 randomly turn into inoperable state about every hundredth cycle (i.e. with the known frequency 1/100T). At the moment of recovery all the degraded instances A1 and A2 turn into the initial operable state. Majority elements are not affected by soft errors, since they do not have memory elements, of which the state can be distorted. However, they can be the source of short false pulses (with the known frequency 1/1000T) that, in turn, can cause soft errors in A1, A2.

The simulation model for this example has the following visual representation (Figure 2):

In Figure 2, individual graphs included in the simulation model are shown with dotted lines. Same-type graphs with identical structure (triplexed modules A1, A2) are grouped with the dual dotted line. In each graph, the thin contour circles designate operable states, the thick contour circles designate the inoperable states. The transitions between states are shown with arrows that connect the states. An arrow entering a transition designates the condition of such transition. It may be a transition that occurred in the current graph or a transition in another graph or Global Event Generator events (wide arrow). If a transition does not have incoming arrows, that means it only depends on the current state in the graph and is not governed by any external events.

Let us examine the model's operating principle using the example of the A1 unit. Primitive graph V1 that has the only state OK simulates the operation of voter V1 that is the source of error V1Fault that affects both the module A1 and A2. This transition, in turn, generates the event A1Fault. In graph O1 that simulates the state of the output line O1, transition to state DIST occurs, indicating that one of the triplexed instances of A1 is inoperable. Let us note that after this one of the instances of graph A1 that was affected by an error retains the inoperable state ERR and stops being the source of errors for O1. The occurrence of the recovery event REC transfers both the inoperable instance A1, and O1 into the state OK. The error in O1 will occur only if the event A1Fault occurs twice over the recovery period, in other words, if two dif-

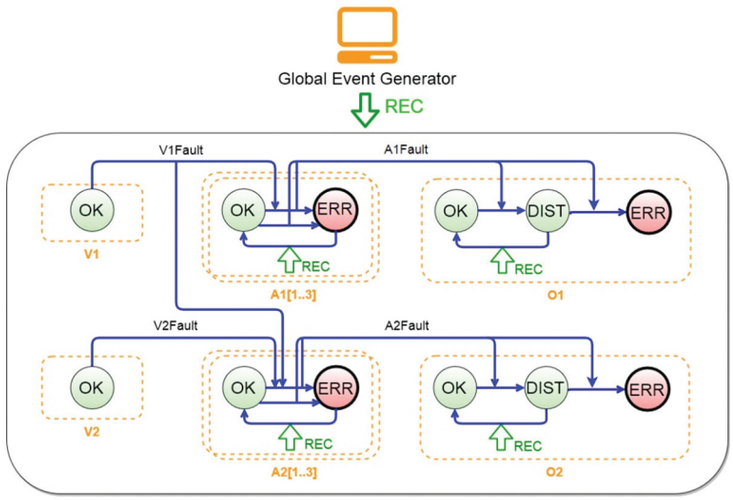

Figure 2. Visual representation of the simulation model

ferent instances of module A1 become inoperable during the recovery period. The event model for module A2 is built in the same way.

Obtaining numerical estimations requires specifying in the model the parameters of transitions and global events. In the text description of model (in the xml format) this is done as follows:

```
   <topLevelDescription>
      <globalEvents>
          <event eventName="REC" distribu-
tionType=
             "CONSTANT" intensity="1.0"/>
      </globalEvents>
      <healthFunction parameters="O1,O2"/>
      <graphs>
          <graph filePath="voter1.gr"
           graphName="V1"/>
          <graph filePath="voter2.gr"
          graphName="V2"/>
          <graph filePath="A1.gr"
          graphName="A1[1..3]"/>
          <graph filePath="A2.gr"
          graphName="A2[1..3]"/>
          <graph filePath="O1.gr"
          graphName="O1"/>
          <graph filePath="O2.gr"
          graphName="O2"/>
      </graphs>
  </topLevelDescription>
```

The globalEvents section describes the events generated by the global events generator. Each event (this applies not only to global events, but also to transitions in the graph) is defined by an "event" record that contains the following parameters:

• eventName, the name of the event in the model;

• distributionType, the distribution law of the random event of the moment of occurrence (CONSTANT, determined with specified frequency, EXPONENTIAL, exponential with specified intensity, GAUSSIAN, normal with specified intensity);

• intensity, specifies the intensity of the occurrence of the event distributed over the exponential and normal distribution laws. For deterministic events, the period between events is fixed.

HealthFunction defined the operability function. In its only parameter (parameters), separated by commas, are given the names of graphs that must be operable in order for the system to be deemed operable.

The graphs section specifies the list of graphs included in the simulation model. To each graph corresponds a record of the type graph with filePath parameters (path to the xml file that contains the graph description) and graphName (name of the graph). If a model contains several identical graphs (in the example at hand those are triplexed modules A1 and A2), structures of the type A1[1..3] can be specified as graph name, as the result of which the model will use 3 graphs with the names A1[1], A[2], A[3].

Given the fault parameters used in this example (soft errors rate of the A1 and A2 modules equals 1/100T), and taking the modules' operating cycle as the measurement unit, the description of graph A1 is as follows:

```
<description>
    <states>
        <state name="OK" isfail="false"
         initialProbability="1.0"/>
        <state name="ERR" isfail="true"
         initialProbability="0.0"/>
    </states>
    <links>
        <link firstNode="ERR"
         lastNode="OK" eventName="REC"/>
        <link firstNode="OK" lastNode=
         "ERR" eventName="V11Fault"
         generateBefore="A1Fault"/>
        <link firstNode="OK" lastNode=
         "ERR" intensity="0.01"
         distributionType="EXPONENTIAL"
         generateBefore="A1Fault"/>
    </links>
</description>
```

The description consists of a list of graph states and links. Each state has a name, an indication of operability (isFail) and probability of the graph being in this state at the start of simulation (sum of these probabilities for all graph states must be equal to 1). Each link has the same parameters as the Global Events Generator events. Additionally, outgoing (firstNode) and incoming (lastNode) states and the name of the external event that is additionally generated at the moment of this transition (in the generateBefore parameter, if the external event must be generated before the transition in the current graph, or in the generateAfter parameter, if the external event must occur after transition).The names of the states and events used in the description of the model correspond to those used on Figure 2.

By launching a calculation procedure for 1000 experiments we obtain the following result (in device operation cycles):

Mean time to failure = **1202.5 (cycles);**

Result error: **± 37.3 (cycles).**

Thus, the mean time to failure was estimated of a computer system unit that consists of recoverable structural blocks. The comparison of the quality of the results of dependability analysis of recoverable units obtained using a simulation model and conventional methods of dependability analysis is examined in [12].

The considered simulation model is applicable for the assessment of recoverable units with complex behaviour and recovery. It is incorporated into the Digitek Reliability Analyzer dependability analysis software [19]. At the same time, beside such units, a hardware and software system includes base blocks, such as batteries, clock speed generators, fuses, etc. The evaluation of such elements' effect on the dependability is more easily done

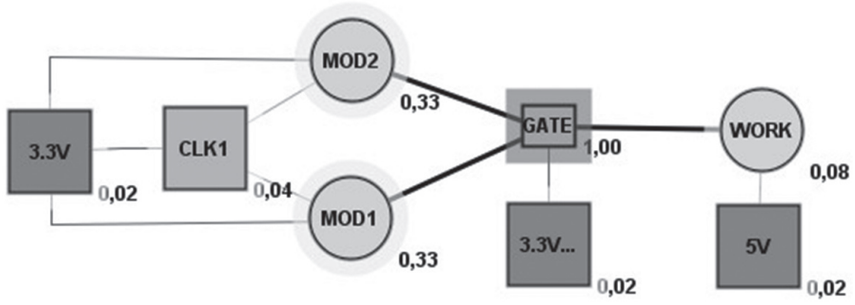

Figure 3. Structural diagram of information system dependability

by conventional methods. For that reason, when Digitek Reliability Analyzer is used, it is suggested analyzing complex recoverable units with the aid of a simulation model, then using a higher level dependability model that contains both the basic hardware elements and complex recoverable units represented as a "black box". The input parameters are the dependability characteristics of reliability previously calculated using the simulation model. Dependability calculation using the upper level model in Digitek Reliability Analyzer is performed logical and probabilistic methods and allows obtaining accurate analytical estimates.

As an example, let us examine a system containing a redundant recoverable unit (Figure 1), of which the dependability was evaluated above using a simulation model, and the connected hardware units. The structural diagram of the device's dependability developed in Digitek Reliability Analyzer is as follows (Figure 3).

The structural diagram (Figure 3) contains two instances of the previously analyzed recoverable unit (MOD1, MOD2) with connected power supply (3.3V) and system clock generator (CLK1). Outputs MOD1 and MOD2 are connected to the switch GATE that ensures correct data setting of the destination workstation (WORK) as long as at least one of the modules MOD1, MOD2 operates. The operation of the GATE element required a power supply (3.3 V...). The system is considered operable as long as workstation WORK operates, which requires the availability of undistorted data in the data line from the switch (GATE), operability of the 5 V power supply and absence of own internal failures.

For each of the elements of the structural diagram parameters of its own internal failures are set. For the elements MOD1 and MOD2 values re used that were calculated using a simulation model (mean time to failure of MOD1 and MOD2 equals 1202.5 cycles). The dependability of the system clock generator and power sources can be found in the respective technical specifications. Next, using DigitekReliability Analyzer the probability function of system operability $P(t)$ is automatically calculated. Its graph is shown in Figure 4 (time t is expressed in the number of cycles of modules MOD1, MOD2).

The vertical line in Figure 4 shows the mean time to system failure (approximately 725 cycles). In order to evaluate the contribution of individual components to this value, the software measures the structural significance of each of them. It is shown next to the right lower corner of the component (Figure 3), lies within the range from 0 (most insignificant components in terms of dependability) to 1 (most significant components in terms of dependability) and depends on the current time and input characteristics of units dependability. The greater is the value of structural significance, the greater "increase" in system dependability is ensured by the growth of such unit's dependability. For the example under consideration

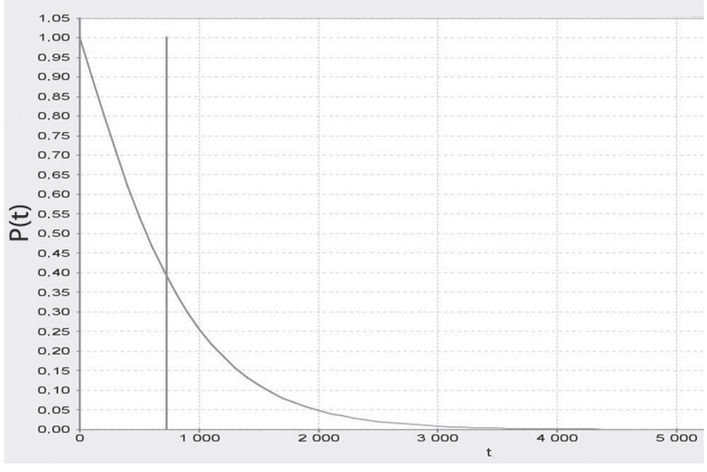

Figure 4. Calculated operability function of information system

(Figure 3) the switch (GATE) has the highest structural significance equal to 1. The recoverable modules MOD1, MOD2 have a structural significance three times smaller (0.33). This means that in order to further improve the dependability of the system, most likely, the fault tolerance of the switch should be increased first. After that, the dependability characteristics must be recalculated and the result evaluated: the mean time to failure must grow, while the structural significance of the units redistribute (the switch will cease to be the most important element). This information enables the designer to evaluate the quality of the current solution and choose the further direction to improved dependability through modification of system architecture.

Conclusion

The simulation model of structurally-complex systems dependability developed by the author enables automated evaluation of the dependability of recoverable hardware and software systems with complex operation algorithms. Its application is especially relevant in the process of design of information management systems that operate under conditions of regular soft errors (e.g. due to adverse radiation conditions).

The developed simulation model allows describing the system's reactions to random events, failures (non-recoverable and recoverable) in its components, as well as non-random events that occur in accordance with the computational algorithm or as the result of operation of the built-in self-repair mechanisms. The simulation model has a sufficient level of abstraction for the description of a wide range of systems. At the same time, its storage format allows developing user representations of the model that are more convenient for system designers.

The use of the simulation model for dependability evaluation of the most complex units alongside well-known analytical methods for dependability analysis of the overall system structure allows facilitating the design of highly dependable radiation resistant systems by incrementally providing the system developer with information required for the selection of the best architecture that meets the specified dependability requirements.

References

[1]. Edmonds DL, Barnes CE, Scheick LZ. An introduction to space radiation effects on microelectronics. Pasadena, USA: NASA, Jet propulsion laboratory, California institute of technology; 2000.

[2]. Amusan OA et al. Single event upsets in deep-submicrometer technologies due to charge sharing. IEEE Transactions on Device and Materials Reliability 2008;8(3):582-589.

[3]. Zhadnov VV, Artyukhova MA. Forecasting spacecraft onboard equipment dependability indicators under low-intensity ionizing radiation. Dependability 2015;1(52):19-24.

[4]. Bochkov KA, Komnatny DV. Mechanisms and probabilities of functional failures of microelectronic elements base under electromagnetic pulse interference. Dependability 2015;3(54):69-72.

[5]. Rollins N et al. Evaluating TMR Techniques in the Presence of Single Event Upsets. Washington DC (USA); 2003. P. 1-5.

[6]. Egorov IV, Melekhin VF. Analysis of Radiation Resistance Improvement Issue for Information and Control Systems at the Stage of Functional and Logical Design. Informatsionno-upravliaiushchiesistemy 2016;1(80):26-31.

[7]. Egorov IV, Melekhin VF. Analysis of Processes in a Finite State Machine under Radiation. Probabilistic Assessment of Information Distortion. Informatsionno-upravliaiushchiesistemy 2016;3(82):24-33.

[8]. Egorov IV, Melekhin VF. Analysis of Reliability and Structural Complexity for Various Implementations of a Finite State Machine Resistant to Soft Failures. Informatsionno-upravliaiushchiesistemy 2017;3(88):34-46.

[9]. Maksimenko SL. Design methodology for embedded systems with built-in self-recovery. Humanities and Science University Journal 2014;8:144-153.

[10]. Maximenko SL, Melekhin VF, Filippov AS. Analysis of the Problem of Radiation-Tolerant Information and Control-Systems Implementation. Informatsionno-upravliaiushchiesistemy 2012;2(57):18-25.

[11]. Egorov IV, Melekhin VF. Methods and Tools for Structural Block Reliability Analysis with Reservation and Periodic Information Recovery at Various Stages of Computing System Design. Informatsionno-upravliaiushchiesistemy 2016;2(81):26-34.

[12]. Maximenko SL, Melekhin VF. Analysis of Reliability of Functional Nodes of Digital VLSI Circuits with Structural Redundancy and Periodic Operational State Recovery. Informatsionno-upravliaiushchiesistemy 2013;2(63):18-23.

[13]. Glukhikh MI. Raschet pokazateley nadezhnosti po modeli struktury vychislitel'noy sistemy [Calculation of dependability indicators based on computer system's structural model]. In. Senichenkov YuB, editor. Vychislitel'nye, izmeritel'nye i upravlyayushchie sistemy: sbornik nauchnykh trudov [Computer, measurement and control systems: collection of studies]. Saint Petersburg: SPbSTU. P. 57-64 [in Russian].

[14]. Cherkesov GN. Nadezhnostapparatno-programmnykhkompleksov: Ouchebnoieposobie [Dependability of hardware and software systems:a study guide]. Saint Petersburg: Piter; 2005 [In Russian].

[15]. Pereguda AI. Mathematical model of dependability for a complex "facility of protection – Safety system" in case of fuzzy initial information. Dependability 2014;1(48):114-128.

[16]. Cherkesov GN, Mozhaiev AS. Logiko-veroyatnostnye metody rascheta nadezhnosti strukturno-slozhnykh sistem [Logical and probabilistic methods of dependability calculation of structurally complex systems]. Kachestvo in nadiozhnost izdeliy [Quality and dependability of products] 1991;3(15):3-75 [in Russian].

[17]. Khakhulin GF, TitovYuP. Simulation model of military aircraft dependable structure and its use in the research of after-sale service processes. Dependability 2014;3(50):16-26.

[18]. Digitek Labs. Reliability Analysis, <http://www.digiteklabs.ru/en/research/relanalyzer>; 2011 [accessed 17.03.2018].

About the author

Igor V. Egorov, graduate student, Department of Computer Systems and Software Technologies, Institute of Computer Science and Technology, St. Petersburg State Polytechnic University, Saint Petersburg, Russia, e-mail: ig-ego@mail.ru

Received on: 03.04.2018