Analysis Of Modeling Methods Of Computer Engineering Digital Devices

V.D. Dmitrienko[1], A.Yu. Zakovorotnyi[2], S.Yu. Leonov[3]

[1]Doctor of Technical Sciences, Professor of Computer Engineering and Programming Department, National Technical University "Kharkiv Polytechnic Institute",
[2]Doctor of Technical Sciences, Professor of Computer Engineering and Programming Department, National Technical University "Kharkiv Polytechnic Institute",
[3]Doctor of Technical Sciences, Professor of Computer Engineering and Programming Department, National Technical University "Kharkiv Polytechnic Institute"

Article History: Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 20 April 2021

Abstract: When designing new digital instruments and devices, there are a vast variety of reasons why the finally designed devices will malfunction. To decrease the number of such failures and to increase design accuracy, various methods and systems for modeling digital devices are used. In these systems, various methods for describing signals in models of designed devices can be used. In this case, three-valued, four-valued, ..., nine-valued, thirteen-valued, as well as analog signal descriptions can be applied. Increasing signal and element models complexity in digital devices allows designing more accurate models. However, when modeling digital devices, multi-valued alphabets do not allow to increase the accuracy of modeling and research of dynamic processes in devices. This is due to the impossibility of taking into account processes and interference caused by both stray capacitances and inductances between separate components of the devices and conductors connecting them, as well as dynamic processes, caused by external electromagnetic fields affecting the device designed. Describing such processes using continuous or *K*-valued differential equations improves the accuracy of digital devices modeling. Nevertheless, the problems of automated testing of these devices and the automation of determining their performance remain unsolved. For automated recognition of failures in the designed digital devices, neural networks, in particular, adaptive resonant theory (ART) neural networks, can be applied, since they have an important property, the ability to retrain when additional information about failures occurs. However, neural networks also have an essential drawback: they do not allow getting more than one solution, although with *K*-valued differential calculus of digital devices, this can occur quite often, which makes it possible to recognize failures that can be attributed simultaneously to two or more different classes of errors, and, therefore, to recognize failures, which can be simultaneously assigned to two or more different classes, and consequently, get more accurate results. In this regard, it is necessary to develop neural networks that could recognize two or more possible solutions (or types of failures). This would expand the field of failures automated detection in the designed digital devices and determine the performance accuracy. Figs.: 3. Refs: 12 titles.

Keywords: dynamic processes; neural networks; *K*-valued differential calculus; digital devices.

1. Introduction

In developing new digital devices, there are a large number of spots where errors can be made for various reasons. Therefore, a thorough designed device modeling is extremely important.

Nowadays, designing new computing devices, a large number of different modeling methods and systems are used to check the product performance. The models of the devices considered are classified according to a number of characteristics: type and level of model description (system, functional and logical, circuitry, component, for production design and, in particular, for printing units design [1 – 7], various value mathematical description of signals and models (Boolean, ternary, four-valued, nine- and thirteen-valued, analog), transient accounting (static and dynamic), etc. The variety of modeling systems and the models used in them is due to two opposite trends in the development of computer engineering digital devices modeling systems. On the one hand, this is a further progressive evolution of universal systems for modeling and designing, oriented to modern supercomputers and time-consuming for their mastering. On the other hand, that is the development of software for personal computers with a user-friendly and intuitive interface for design and studies of some relatively narrow classes of digital devices, or devices operating in conditions that do not require time-consuming consideration of some factors ensuring the reliable operation of its modules at the target frequency. It is obvious that the development of each class of modeling systems is relevant, especially taking into account that the development and modernization of universal modeling systems is very time-consuming, and the period of their use is short due to the hardware and system software of supercomputers.

Although binary, ternary and four-valued description of digital signals in designed devices is used when modeling discrete devices, and requires only one clock time to evaluate their performance, such signal descriptions do not allow to identify the dynamic risks of failures during the transition from one stable state to

another, static risks of failures in stable states "0", "1", etc. Concerning this fact, for more detailed studies of digital circuits functioning, multi-valued Fantozi alphabets were introduced [4], which require at least four clock times to evaluate the digital circuits performance. This, in turn, requires the introduction of variables and alphabets defined at several clock times of digital devices performance, which leads to the increase in the number of characters used in such alphabets. In work [2], *A*-alphabet is described, containing 23 characters. Such alphabets, on the one hand, make it possible to describe the digital circuits operation in more detail than binary alphabets, and, on the other hand, they significantly complicate the modeling of discrete devices developed. Moreover, existing and perspective multi-valued alphabets do not allow describing and modeling digital devices operation taking into account interference, caused by stray capacitances and inductances between separate components of the devices and conductors connecting them. In such cases, it is necessary to use systems of equations as models, which consist of two parts: subsystems of logic algebra equations that describe the functioning of the developed devices, and systems of ordinary differential equations, which allow the calculation and determination of stray interference. Since the solution of such equation systems is very time-consuming, despite the relevance of accounting for stray differential connections when modeling computing devices, these models are rarely used in practice.

The necessity of taking into account stray connections between the components of digital devices and the inability to use ordinary differential equations for their description due to time-consuming modeling has led to the development of a new class of models – *K*-valued differential equations [8, 9]. These models do not require such excessive calculations as models based on systems of continuous ordinary differential equations, and model any processes that are described using multi-valued alphabets, in particular, Fantozi alphabet or *A*-alphabet. Furthermore, in addition to the qualitative evaluation that multi-valued alphabets can provide, *K*-valued differential models allow evaluating the duration, amplitude and shape of the pulses at the inputs and outputs of elements and blocks under research. As an example, we may consider modeling with the thirteen-valued Fantozi alphabet and a system of *K*-valued differential equations of device functioning shown in Fig. 1.

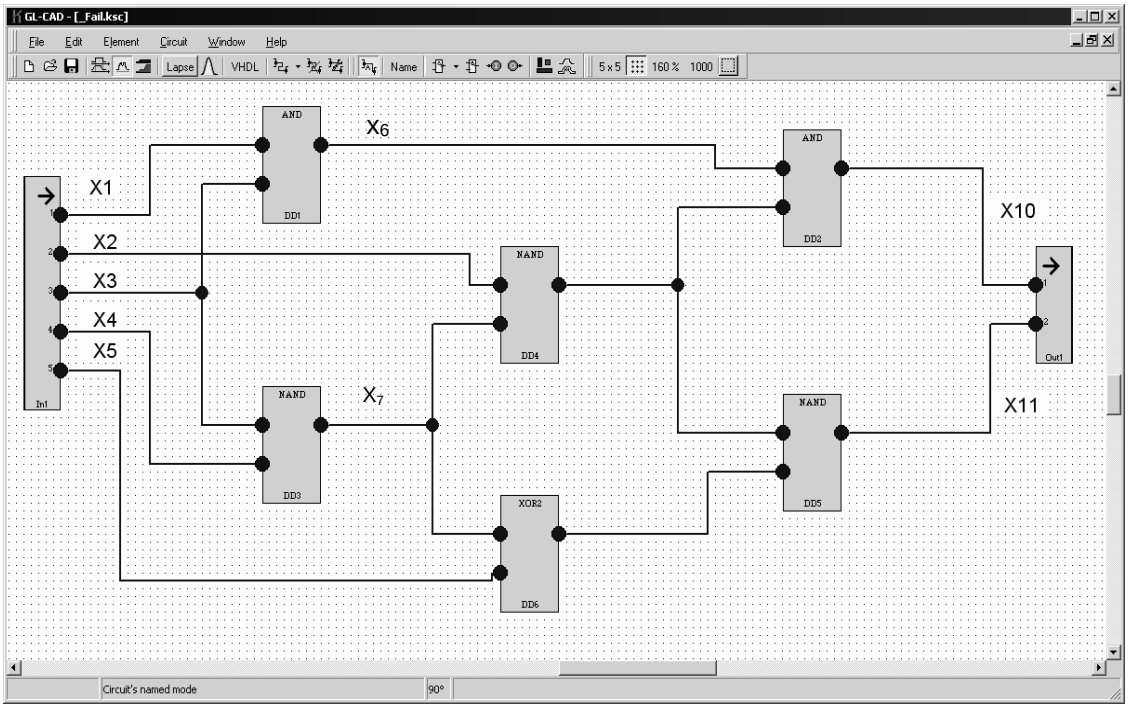

Fig. 1. Device structure in *K*-valued modeling system

The main component of such a device is the "AND" element the states of which can be presented in the 13-valued alphabet Fantozi as follows.

"0" – signal corresponding to a static signal of "0" level;

"1" – signal corresponding to the level of a static "1";

"*X*" – uncertainty;

"*E*" – smooth transition signal from "0" to "1";

"*H*" – smooth transition from "1" to "0";
"*P*" – static risk of failure at "0";
"*V*" – static risk of failure at "1";
"*F*" – dynamic risk of failure in the transition from "0" to "1";
"*L*" – dynamic risk of failure in the transition from "1" to "0";
"O" – transition from uncertainty to "0";
"*I*" – transition from uncertainty to "1";
"*A*" – transition from "0" to uncertainty;
"*B*" – transition from "1" to uncertainty.

The elements of the 13-valued Fantozi alphabet are depicted in Fig. 2, which is obtained in the system of *K*-valued modeling while $K = 7$.

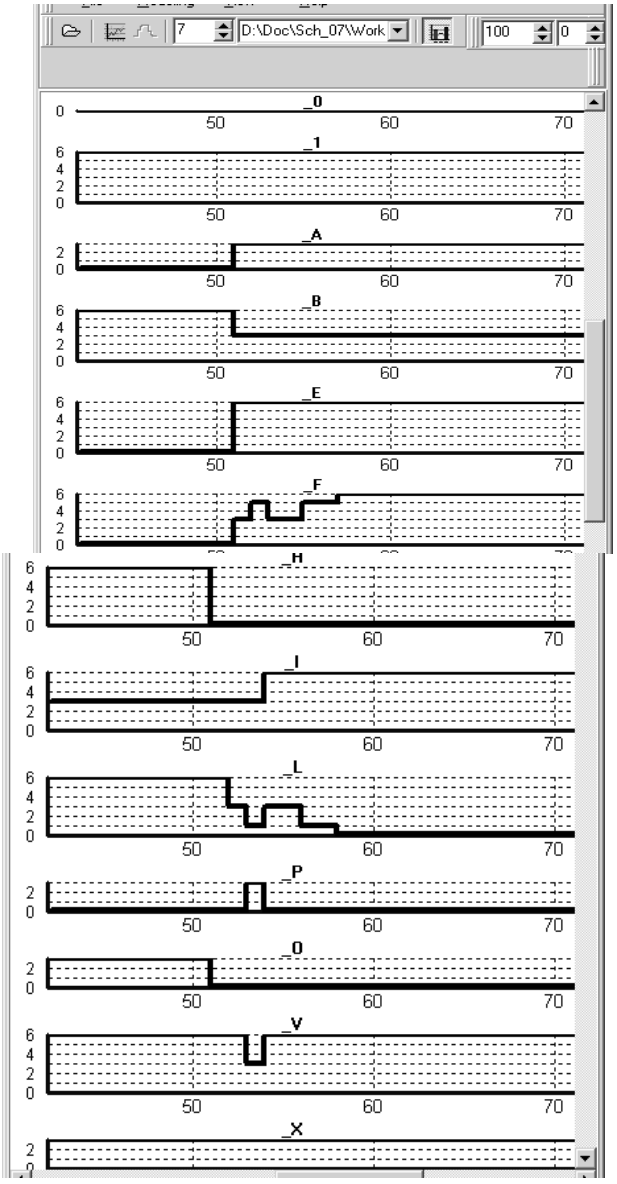

Fig. 2. Elements of the 13-valued alphabet, obtained when studying digital devices in the *K*-valued modeling system while $K = 7$

In the upper left of the figure, the signals "_0" and "_1" correspond to static signals of "0" level and level of static "1". Further, "*_B*" corresponds to the transition from "1" to uncertainty. Signals "*A*", "*_E*", "*_F*" model the transition from "0" to uncertainty, to "1" and dynamic risk of failure in the transition from "0" to "1", respectively. Signals "*_H*", "*_L*" on the right of the figure correspond to a smooth transition from "1" to "0" and the dynamic risk of failure in the transition from "1" to "0". Signals "*_I*", "*_O*" correspond to the transition from

uncertainty to "1" and "0", respectively. With a signal "*_X*" uncertainty is modeled, and signals "*_P*", "*_V*" are used for modeling static risk of failure at "1" and "0", respectively.

The results of modeling the circuit in Fig. 1 are represented in Fig. 3. Signals are sent to the circuit inputs *X*1, …, *X*5, while X2 = X4 = "1", X5 = "0", and signals X1, X3 are in a state of uncertainty. Therefore, at the output of the DD1 element, output signal "1" instantly cannot appear. On the one hand, the DD1 element has a delay (7 time ticks); therefore, the uncertainty signal will appear only after the seventh clock time. On the other hand, the DD1 element is defined after the power analysis, taking into account the accumulated power at its input, necessary for transition of the DD1 element, this element toggles. Counting of the accumulated power *E* is calculated by the ratio

$$E = 2\int_{t_k}^{t_k+T_1}\frac{U^2(t_i)}{R_0}dt_i = 2\sum_{t_i=t_k}^{t_k+T_1}\frac{U^2(t_i)}{R_0}h, \quad (1)$$

where 2 is the coefficient concerning the number of inputs of the DD1 element; t_k is the start of interval time for accumulating power, required for transition of the element; T_1 is the interval time for accumulating power, required for transition of the element; R_0 is the input resistance of logical element; h is a sampling step. Taking into consideration the response of power *E*, the model can be obtained and calculated for any digital element of the designed device. After accumulating power, the "AND" element of DD1 firstly, at the 17th clock time, transits into state $U_{out\,\mathrm{DD1}}$ = 5, and then, at the 19th clock time, it transits into "$U_{outDD1} = 6$".

At the 51st clock time, the input signals X1 and X2 take "0" value and at the output of the "AND" element with a time delay of the DD1 element, "0" signal appears. Similarly, the signals of all circuit elements in Fig. 1 can be calculated.

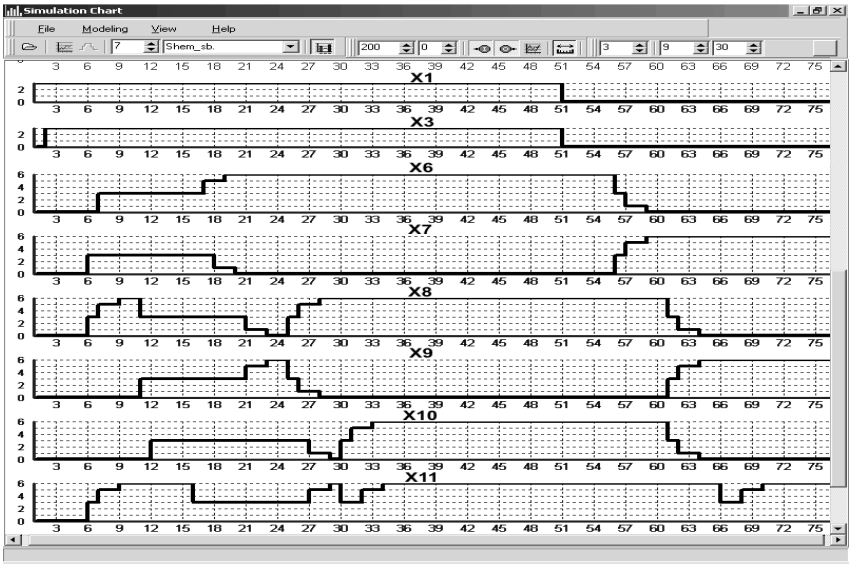

Fig. 3. Timing diagrams of the circuit functioning in Fig. 1

It should be outlined that it is important not only to describe the functioning processes of the designed digital devices mathematically correctly and to model them using the developed modeling system, but also to automate the process of determining failures in these products during their research and testing. Such automation reduces the role of the "human factor" in products fine-tuning and determining their performance, and saves time significantly to receive normally functioning digital devices. To automate these processes for a modeling system based on *K*-valued differential calculations, a number of *K*-valued neural networks have been developed, allowing classifying adequately the signals at the outputs of the designed products using the 13-valued Fantozi alphabet and various quantization of the digital signals amplitude. The influence of the *K*-state number of discrete neurons in *K*-valued neural network was taken into account: 5, 7, 11, 13. Since the software was designed in such a way that to change the value of modeling, it was enough to set only the value *K*, this allowed modeling a digital circuit with various *K* values without any difficulties. At the same time, the only limitation took place: *K* should be a prime number [8, 9]. The correct or incorrect functioning of computing devices was

determined with thirteen types of signals (Fig. 2) based on Fantozi alphabet. After studies of various neural networks, the neural network of adaptive resonance theory ART-1 was chosen as basic [9, 10]. It can retrain, i.e. remember new classes of images without losing information already stored. To work with *K*-valued input signals, binary neurons in the ART-1 network were replaced by *K*-valued ones. Algorithms for the network functioning with *K*-valued elements were adjusted. As a result, ART-1K neural network was obtained, which classified correctly up to 95-97% of input signals using Fantozi alphabet. On the one hand, this is a good result for a system of recognition or classification of input information. On the other hand, the percentage of correct classification is still unsatisfactory to develop a system for automated testing of digital devices, as it is rather time-consuming for specialists, who have to analyze situations individually, while automated testing systems are inoperative. This could happen due to the following reasons:

1. The absence of appropriate signals corresponding to the input signal in the set of Fantozi alphabet elements.
2. The ambiguity of the neural network definition of Fantozi alphabet element.
3. The lack of a system for assigning and fixing unrecognized input processes.
4. Dependence of test results on the neural network training process.

The purpose of these studies is to develop a new *K*-valued neural network of the adaptive resonant theory in order to improve functioning quality of the digital products testing system, developed with *K*-valued differential calculations and multi-valued alphabets.

The *K*-valued neural network of the adaptive resonant theory is given in Fig. 4, allowing to solve the task in many ways.

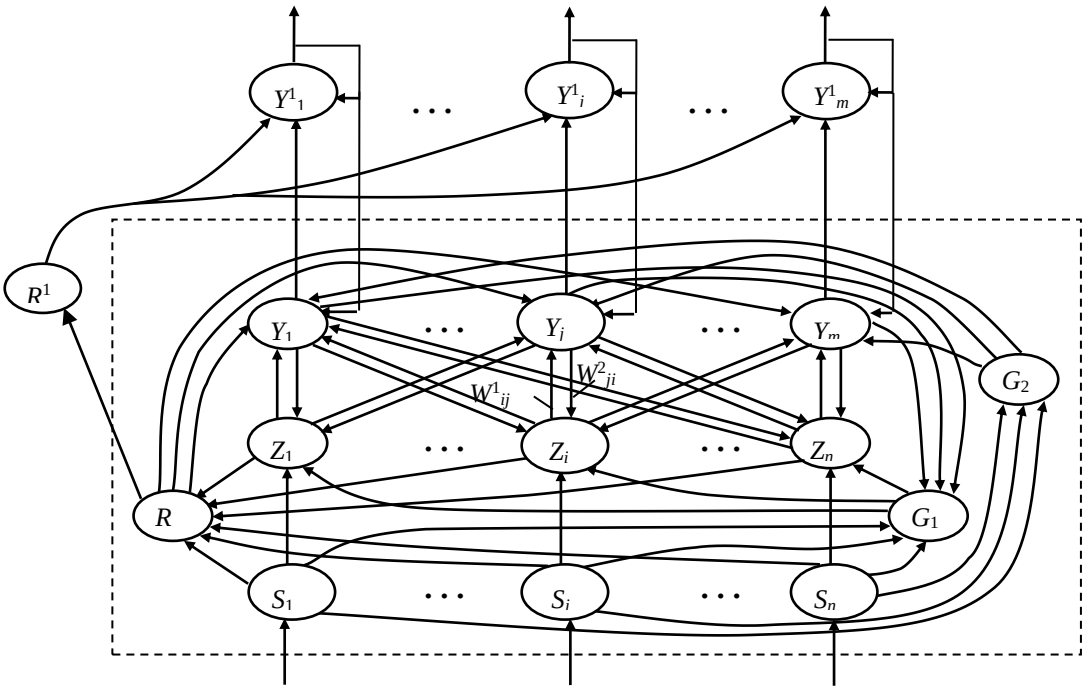

Fig. 4. ART discrete neural network architecture to improve testing efficiency of *K*-valued models

The dashed rectangle shows the architecture of the ART-1K neural network. Its architecture is the same as that of a neural network ART-1, in detail described in the works [9, 12].

The network architecture includes three groups of neurons: field F_1 of input processing neurons, consisting of two layers of *S*- and *Z*-elements, the layer of recognition of *Y*-neurons and control neurons *R*, G_1, G_2 (Fig. 3).

Field F_1 of input processing neurons consists of two layers: input layer of *S*-elements and interface layer of *Z*-elements. The input layer perceives the presented images and transmits the received information to neurons of interface *Z*-layer and control neurons *R*, G_1, G_2. Each element Z_i ($i = 1, \ldots, n$) of interface layer is connected with

each element Y_j (j = 1, …, m) of recognition layer Y by two types of weight links. Signals from the interface layer are transmitted to layer Y by upwards links with weight W_{ij}^1, and from the recognition layer to the interface by links with weight W_{ji}^2, (j = 1, …, m, i = 1, …, n). Due to a large number of links in Fig. 3, only one pair of links with weight W_{ij}^1, W_{ji}^2 between interface and recognition layers of elements is demonstrated.

Layer Y is a layer of competing neurons. At any time, each element Y_j (j = 1, ..., m) of the recognition layer is in one of three states:

– active (output signal U_{outYj} of neuron Y_j is equal to d: $U_{outYj} = d$; d = 6 for ART-1K with the value of the input alphabet K = 7;

– inactive (U_{outYj} =0, but neuron can participate in the competition);

– delaying (U_{outYj} = –1 and the neuron is not allowed to compete upon presentation of the current input image).

After presenting the input image, only one recognition neuron remains active, other Y-elements have "0" or negative output signals. An assigned recognition neuron in the training mode is allowed to be trained by the input image only if its weight vector of links from layer Y to layer Z is similar to input vector. This decision is made using R-neuron and a special parameter, i.e. a similarity parameter, and signals, transmitted from the input and interface layers of elements. Through auxiliary elements, training or delaying (reset) of an assigned recognition Y-element takes place, with further quitting the competition with repeated presentations of the same input image, when new candidates for input image training are assigned from Y-layer.

Most of the links given in Fig. 3 are exciting the input layer of S-elements to neurons of R, G_1, G_2 and Z-layer; neurons G_1, G_2 to neurons of layer Z and Y, respectively. Delaying signals transmit only multiple links from interface elements to R-neuron and from Y-neurons to element G_1, from R-neuron to the winner neuron in the recognition layer. All ART-1K network links transmit K-valued signals based on the alphabet M = {0, 1, 2, …, K – 1}.

Each element in an interface or Y-layer of ART-1K network has three input sources. Random interface element Z_i (i = 1, …, n) can receive signals from an element S_i of the input layer, from elements of Y-layer and from the neuron G_1. Similar element Y_j (j = 1, …, m) can receive signals from interface elements, neurons R and G_2. To transmit neurons of the interface or recognition layers into an active state, two sources of input exciting signals are required. Since each of the neurons considered has three possible signal sources, the condition for the excitation of these neurons is called the two-of-three rule.

In the initial state neurons R, G_1, G_2 and some of input layer S have "0" output signals. When got to the inputs of S-elements of K-valued component of the presented image, part of them, receiving non-zero input signals, transmits into an excited state ($U_{\text{out}} > 0$). Exciting signals from the outputs of these neurons transmit neurons G_1, $G_2$to the state "6", and also get to the inputs of the corresponding neurons of the interface layer. Interface layer neurons receiving signals from input layer neurons and element G_1, following the two-of-three rule, transmit into an active state and send their exciting signals through links with the weight W_{ij}^1 (i = 1, …, n, j = 1, …, m) to inputs of neurons Y_j of the recognition layer. The neurons of the recognition layer become active according to two-of-three rule, receiving exciting signals not only from elements of the interface layer, but also from the element G_2. Output signals of active Y-neurons are determined by the ratio

$$U_{\text{out}\,Y_j} = U_{\text{in}pY_j} = \sum_{i=1}^{n} W_{ij}^1 U_{\text{out}Z_i}, \quad j = 1, ..., m,$$

under the condition

$$0 < U_{outYj} \le 6.$$

Then, the lateral process of assigning "1" element J with the highest output signal takes place in Y-layer of neurons. All neurons of Y-layer, except the winner Y_J, transmit in an inactive state "0" (U_{outYj} =0), and the winner neuron transmits in a state with "1" (K = 6) output signal. The signal of the winner Y-neuron delays the control neuron G_1, and also transmits through the links with weight W_{ji}^2 to the inputs of neurons of the interface layer. As the elements of the interface layer follow the two-of-three rule, with the absence of an exciting signal from a neuron G_1 in the active state, only those interface elements, which receive signals from both the input layer element and the winner Y_j-neuron of the recognition layer, will participate. Delaying signals of the active elements of the interface layer get to the inputs of R-element, which receives exciting signals from neurons of the input layer. Depending on the values ratio of the exciting and delaying signals, the output signal of the control

element *R* is determined. With "0" output signal of *R*-element, there is a resonance in the network, and the links weight of the winner *Y*-neuron are trained; when the output signal is "1", the winner *Y*-neuron delays (U_{outYj} = – 1) and actually loses the opportunity to participate in the competition upon presentation of the current image. Then, another winner neuron is selected in *Y*-layer. If the input image is not enough alike any memorized one, all distributed *Y*-neurons eventually delay, and an unallocated *Y*-element becomes the winner and remembers a new image in its weight.

ART-1K neural network training algorithm is similar to ART-1 neural network training algorithm [10, 11]. The difference between the two algorithms is in the use of different neurons: in ART-1K network, *K*-valued neurons are applied, and in ART-1 network binary ones are applied.

The 13-valued Fantozi alphabet takes into account many peculiarities of digital device modeling, but not all. For example, the alphabet contains signals of uncertainty and static "0", but it cannot be used to determine an input signal that changes from "0" to uncertainty level. In this case, there are two solutions: "_0" signal and "_X" signal. The layer of neurons Y^1 is introduced to identify those signals that may relate to two or more elements of the Fantozi alphabet. These neurons also follow the two-of-three rule. The layer of these neurons does not require training, and in the recognition mode they are triggered by a pair of signals: the winner neuron from the *Y*-neuron layer and the output signal of the control neuron R_1, which inverts the output signal of the neuron R. The output signal of the triggered neuron of the Y^1-neurons layer fixes its output signal and delays the corresponding neuron in the layer of *Y*-elements. Then, the search for a new Y-neuron starts, which contains in its memory information about the value of the similarity parameter corresponding to another element of Fantozi alphabet. Thus, two or more elements of Fantozi alphabet are selected that correspond to the input signal. If the input signal does not correspond to one element of Fantozi alphabet by the value of the similarity parameter, the outputs of the Y^1-layer neurons will be "0" signals. The output signals of the Y^1-layer neurons are summed using $\sum$. If the sum of the signals of the Y^1-layer neurons is "0", the permissive output "0" signal of this neuron gets to the inputs of the S^1-layer neurons, which stores information about the input vector received at the inputs of the S^1-layer neurons. As a result, a vector appears at the output of the S^1-layer neurons, which did not cause a reaction of the neural network and which can be used to retrain this network.

2. Conclusions

The analysis of methods for modeling digital devices using multi-valued alphabets was carried out. Despite the vast capabilities of discrete devices modeling, multi-valued alphabets do not allow to identify failures connected with transients, which may occur during the transition from one stable state, described by a multi-valued alphabet, to another. This approach also does not allow describing and modeling processes connected with stray parameters of elements on boards or in crystals. In this regard, for a more accurate modeling, K-valued differential calculations can be used in the digital devices design, which allow modeling the designed digital devices in more detail than using multi-valued alphabets. It also allows describing and studying processes connected with stray capacitances and inductances at the inputs and outputs of separate digital elements, as well as dynamic processes, caused by external electromagnetic fields. To automate the processes for determining the designed digital devices performance, new networks of adaptive resonance theory are proposed, which allow not only to identify certain types of failures, but also failures of several different classes.

References

1. Bennets R.D. Testable Logic Design / Translation from English Derbunovich L.V. – Moskow.: Radio and Communications, 1990. – 176 p.
2. Khakhanov V.I. Models of digital and microprocessor structures and methods of their analysis in the diagnostic service system. Dis. doctor tech. sciences: 05.13.02 and 05.13.08. – Kharkov: KhTURE, 1996. – 350 p.
3. Jutman A. At-Speed On-Chip Diagnosis of Board-Level Interconnect Faults // Proc. of 9th European Test Symposium (ETS'04). – France. – 2004. – P. 2-7.
4. Chouki Aktouf. A Complete Strategy for Testing an On-Chip Multiprocessor Architecture // IEE Design & Test of Computers. – 2002. – P. 18-28.

5. Malyshev N. New opportunities in the system of functional verification and modeling HDL-projects Delta Design Simtera // Modern electronics, 2019. – No. 8. – P. 64-66.
6. Zyrin I., David Marracchi. Design for manufacturing (DFM). Part 1. Choice of materials // Modern electronics, 2019. – No. 9. – P. 66-71.
7. Zyrin I., David Marracchi. Design for manufacturing (DFM). Part 2. Preparing a PCB design strategy // Modern electronics, 2020. – No. 3. – P. 66-74.
8. Dmitrienko V.D., Leonov S.Yu. *K*-value differential calculus and digital device simulation. – Kharkov: Transport of Ukraine, 1999. – 223 p.
9. Leonov S.Yu. Theory of automated design of electronic devices based on *K*-value modeling. Dis. doctor tech. sciences: 05.13.05. – Kharkov: NTU "KhPI", 2014. – 325 p.
10. Yampolsky L.S. Neurotechnology and neurosystems / Kiev: Monograph. – " Dorado -Printing ", 2015. – 508 p.
11. Noskov V.I., Dmitrienko V.D., Zapolovsky N.I., Leonov S.Yu. Modeling and optimization of locomotive control and monitoring systems. – Kharkov: KhFI " Transport of Ukraine ", 2003. – 248 p.
12. Fausett L. Fundamentals of Neural Networks. Architectures, Algorithms and Applications / L. Fausett. – New Jersey: Prentice Hall International, Inc., 2006 – 483 p.