MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE
KHARKOV NATIONAL UNIVERSITY OF RADIOELECTRONICS

ISBN 966-659-113-8

Proceedings of IEEE East-West Design & Test Workshop (EWDTW’05)

Odessa, Ukraine, September 15 – 19, 2005

CONTENTS

BRAINLIKE COMPUTING

Shabanov-Kushnarenko Yu., Klimushev V., Lukashenko O., Nabatova S., Obrizan V., Protsay N.

Kharkiv National University of Radio Electronics, Lenin ave 14, Kharkiv, 61166, UKRAINE,
obrizan@kture.kharkov.ua

Abstract. This paper offers mathematical foundation of brainlike computer. New approach to making artificial intelligence is discussed: the human intelligence is considered as some material embodiment of the mechanism of logic. Also hardware efficiency of logic net implementation is shown.

Keywords: brain-like computer, predicate algebra, hardware implementation, logic net, language model.

Urgency of research is determined by the necessity of design of the parallel computer for significant performance increase in comparison with software implementation on von Neumann architectures.

The goal of the research – design of the parallel computer operation by principles of a humane brain and designed on modern element base.

To reach the goal it is necessary to solve the following *tasks*: 1) designing the new method of artificial intelligence: the humane intelligence is considered as some material implementation of the mechanism of logic; 2) algebraization of logic; 3) formalization of logic net model; 4) developing logic synthesis procedures for logic net; 5) designing logic net design flow; 6) analysis of hardware implementation efficiency.

Quickly progressing computerization and informatization demand constant increase of productivity of electronic computers. However, it is more and more difficult to do it. Reserves of increasing the speed of computing elements of the computer are getting exhausted. There is a way of escalating a number of simultaneously operating elements in the computer processor. Nowadays there is a practical possibility to build computers with the number of elements up to 10^8, based on successes of microminiaturization and reduction in price of electronic elements and on achievements in the field of automation of design and manufacturing of computers. However, with the present serial computers operation based on the principle of program control by J. von Neumann, it is senseless to do this, as there is only a small number of elements in operation during each period of time in them simultaneously. Attempts of conversion to parallel machines do not provide the expected growth of their productivity. For example, productivity of multiprocessing computers does not grow proportionately to the number of processors available in them as, apparently, it should be, but much slower. There are essential difficulties in attempts of creation of high-efficiency neurocomputers, which are constructed as formal neuron networks. Meanwhile, there is the "computer" created by nature, namely – a human brain for which the problem of high-grade parallelism of information processing is completely solved. Human brain is a slow mover in comparison with the modern computer. Its “clock frequency” can be estimated by throughput of nervous fibers. It is known, that each nervous fiber can pass not more than 10^3 pulses per a second. Through the conductors of modern computers it can be transferred about 10^9 pulses per a second. Hence, the computer surpasses a human brain in terms of speed of work of computing elements in 10^9:10^3=10^6 times. And nevertheless, the brain, due to a parallel principle of action, works faster and is capable to solve immeasurably more difficult tasks, than the most powerful modern computers with program control. It is caused by the fact that the human brain incorporates about 10^{15} computing elements (acted by synapses – interfaces between the ends of nervous fibers), and all of them are operating simultaneously, according to neurophysiology. In serial computers at any moment only the small number of elements operates in parallel.

By most preferential estimations, there are not more than 10^4 elements in it operating simultaneously. Thus, according to the number of elements operating in parallel, the brain surpasses the machine by 10^{15}:10^4=10^{11} times. As a result, the brain surpasses the modern serial computer in productivity by 10^{11}:10^6=10^5 times. If it were possible to create the parallel computer operation by principles of a brain which has 10^{15} elements the machine surpassing a human brain in productivity by 10^9:10^3=10^6 times would created. So, the parallel computer operating by principles of a brain and made on modern element base (*brainlike computer*), according to the above-stated estimations, in case of its creation will surpass the productivity of the present serial computers by 10^{11} times, and a human brain – by 10^5 times.[1]

Why experts on neurocomputers did not manage to design *brainlike computer* with the above productivity until now in spite of the fact that they have been dealing with this problem for about half a century already? Attempt of the answer to this question can be found in D.Hjubel's book “The Eye, the Brain, the Sight” – the Nobel prizewinner, one of the world's largest experts in the field of anatomy and physiology of human brain neural networks. He writes: “Sometimes they say that the nervous system contains huge number of casual interneural connections. Though orderliness of connections is not always obvious … in monstrous complexity of nervous system it is almost always possible to see a known degree of orderliness. …it becomes doubtless, that orderliness serves some purpose. … In biology there is a similar belief in functional validity and finally even in apprehensibility of structures which have not been invented by someone, and were improved during millions years of evolution." [4]. The statement of Hjubel can be understood in the sense that technical neural networks are not at all what their biological prototypes are. While the neurostructures anatomy at a microscopic level is now well investigated, research of physiology of these structures, despite separate achievements, still hampers. Therefore engineers at their own risk build hypotheses about principles of action of neural structures.

During last 40 years the scientific direction – the theory of intellect, which attempts to set going the decision of a problem discussed here, is being developed at Kharkov National University of Radioelectronics. The essence of the approach is that the human intelligence is considered as logic in operation, as some material embodiment of the mechanism of logic. Works on *algebraization of logic* have been executed. As a result special mathematical device for formular representation of relations and operations on them is developed which is referred to *as logic algebra structures*. Relations are interpreted as ideas of intelligence, and operations on them – as thinking. Circuit realization of the formulas describing algebra-logic structures, results in characteristic engineering networks (not used earlier) named *logic networks*. The main thing in the given method is a movement from top to bottom: From the general system reasons to *logic algebra* structures, and from them – to logic net which are further identified with biological neural structures [6]. Experts in neurocomputers nowadays try to move in a different way: from biological neural networks to principles of their operation, and from them – to engineering solutions. This way results in significant difficulties because of lack of knowledge of functions of biological neural structures. So, principles of construction of *brainlike computers*, essentially differ from all those used until now at making parallel processings of the information, in particular – at creation of the parallel computer. Basis of *brainlike computer* is a logic net mentioned above.

The logic net is intended for performance of operations on relations. It is just that device of *brainlike computer* with which help it will think. It is the processor of *brainlike computer*. It is literally for some steps (tens steps) that makes shares of second, the computer gives the answer to the question.

Relations and predicates

What is circuit realization of a logic net and how its work is carried out? To answer the given question it is necessary for the beginning to learn to write down relations and operations on them with the help of formulas. As experience of development of a science and technology shows, there is no more convenient and practical means of the description of objects, than formulas. Formulas provide not only the description by objects, but also express their deep properties and structure. Addressing to experience of mathematics, it is possible to notice, that formulas always express only functions. Probably, no other mathematical objects can be described by formulas. But the relation is not a function, but something more general.

Let's give the expanded characteristic to a concept of relation. Let us choose some nonempty set U which elements we shall name *objectss*. The set U is called *objects universum*. It can be both final, and infinite. Let us choose m nonempty, unessentially various subsets $A_1, A_2, \dots, A_m$ of the universum U. The Cartesian product $S = A_1 \times A_2 \times \dots \times A_m$ of sets $A_1, A_2, \dots, A_m$ is called *object space* S with *coordinate axes* $A_1, A_2, \dots, A_m$ above universum U. Let's consider the set $V = \{x_1, x_2, \dots, x_m\}$ of various variables $x_1, x_2, \dots, x_m$ which are named *object variables* of space S. The set V is called *variables universum* of space S. Values of a variable $x_i (i = \overline{1, m})$ are elements of set A_i, so $x_1 \in A_1$ $x_2 \in A_2$, … $x_m \in A_m$. Sets $A_1, A_2, \dots, A_m$ are called *areas of the definition of variables* $x_1, x_2, \dots, x_m$. Each variable $x_i (i = \overline{1, m})$ corresponds to some fixed area of the definition A_i. The space S can be considered as a set of all vectors of a kind $(x_1, x_2, \dots, x_m)$, each of which satisfies to a condition $x_1 \in A_1$ $x_2 \in A_2$, … $x_m \in A_m$. Any subset P of space S is called *the relation* formed in (or otherwise: defined on) space S [5].

There are different ways of expressing relations: by sets of objects vectors, by graphs, by diagrams, by tables. But among them there is no formular one.

Let's try to find a way of recording relations by formulas. For the decision of the given problem we shall define the concept of a predicate.

Any function $P(x_1, x_2, \dots, x_m) = \xi$ displaying the Cartesian product $A_1 \times A_2 \times \dots \times A_m$ of sets

$A_1, A_2, ..., A_m$ in set $\Sigma = \{0,1\}$ is called *the predicate* defined on the Cartesian product S. Symbols 0 and 1 are called boolean *elements*, Σ - set of all Boolean elements. The variable $\xi \in \{0,1\}$ being value of a predicate P, is called boolean. There is a biunique correspondence between relations and predicates.

Let L be set of all relations on S, M – set of all predicates on S. The relation P from L and a predicate P from M are *corresponding* each other, if for any $x_1 \in A_1$, $x_2 \in A_2$, …, $x_m \in A_m$

$$P(x_1, x_2, ..., x_m) = \begin{cases} 1, \text{if } (x_1, x_2, \ldots, x_m) \in P; \\ 0, \text{if } (x_1, x_2, \ldots, x_m) \notin P. \end{cases} \quad (1)$$

Back transition from a predicate P to the relation P can be carried out with help of the following rule:

$$\text{if } P(x_1, x_2, ..., x_m) = 1,$$
$$\text{then } (x_1, x_2, \ldots, x_m) \in P;$$
$$\text{if } P(x_1, x_2, ..., x_m) = 0, \quad (2)$$

then $(x_1, x_2, \ldots, x_m) \notin P.$

The rules (1) and (2) establish biunique correspondence between all relations of set L and all predicates of set M defined on S.[5]

The predicate $P(x_1, x_2, ..., x_m)$, in difference of the corresponding relation P is a function, therefore there is a hope, that it will manage to be expressed as the formula.

The algebra of predicates

The basic predicates playing a role of initial building blocks, and the basic operations providing connection of blocks in a uniform design what is the required formula will be necessary for construction of formulas for us. In a role of basic predicates we use predicates 0, 1, and also the special predicates named *predicates of recognition object* a *on a variable* x_i $(i = \overline{1,m}, a \in A_i)$ and written down as x_i^a.

The predicate x_i^a «recognizes» some object x_i, chosen from set A_i comparing it with a object a.

$$x_i^a = \begin{cases} 1, \text{if } x_i = a, \\ 0, \text{if } x_i \neq a. \end{cases} \quad (3)$$

The indication of a object a and value of an index *i* completely defines a predicate of a kind (3). In a role of ways of connection of the building blocks we have chosen: predicates 0, 1 and predicates of a kind x_i^a – we use operations of a disjunction, conjunction and denying of predicates. As a result so-called *algebra of predicates* is received.

The language of algebra of predicates is universal, it makes possible to describe the structure of any objects formally. Predecessors of the given algebra are algebra of boolean functions (algebra of logic), created by J. Boulle in XIX century, and multiple-valued logic (the first quarter of XX c, Post) [3]. Functions of algebra of logic accept the same values, as predicates. However arguments of function of the algebra of logic are binary, for the predicates they are alphabetic. Arguments of function of multiple-valued logic accept the same values, as arguments of predicates. However values of functions of multiple-valued logic are alphabetic, and for predicates they are binary.

As we see operations $\vee$ $\wedge$, $\neg$ in a combination to predicates 0, 1 and every possible predicates of a object recognition form sufficient assortment of expressive means for formular record of any predicate of any type, and any relation corresponding to it. Taking into account, that relations are a universal means of description of any objects, we can conclude, that it is possible to translate any dependences known in science into language of formulas of algebra of predicates. In universal language (algebra of predicates namely is such language) it is possible to write down any law of nature, any computer program, any mathematical parity, in general – any idea, absolutely everything, that is possible to observe around and inside us. It is also important that the mechanism of intelligence can not be high-grade described without using language of the algebra of predicates. The intelligence is an universal tool, and it can be covered only by universal language.

We will not receive a high-grade artificial intelligence and computer processing of knowledge, if we do not understand that algebra of predicates for writing down of relations plays the same role, as school algebra in integral calculus for writing down of sub-integral function. Until now it turned out, that we carry out operations on relations, without having means for formal recording relations. A vivid example are databases. The relations are recorded in them as tables, and various operations are carried out with them. However recording the tables by formulas in a DB is not stipulated, for this reason we have no opportunity to build combinational circuits for a DB.

Just as the algebra of Boolean functions in due time was used as a base for combinational circuits for universal computers (it was necessary to add only an element of a delay on 1 step to circuits of the formulas of algebra of boolean functions that the machine had memory), so logic net (circuits which turn out as a result of realization of formulas of the algebra of predicates), are the main element at designing of the brainlike computer. In the algebra of boolean functions the machine algorithm (a procedural way of processing of the information) is applied at a calculation of values of functions. In brainlike computers the declarative principle for data processing is used. The circuit sets connection $P(x_1, x_2, ..., x_m) = 1$. It is necessary for machine not to calculate functions values, but to find sets of roots of the equations. The algebra of predicates forces to change ideology of mathematics: the mathematics for each value of argument a single finds or a multitude of one multiple values of a function. It is due to the fact that in an external world there is a unique value in each point of space, we deal with the certain fact. In the internal world there are not facts, but knowledge of the facts. Therefore in the equation not one value of a known variable is defined, but the set of values, and accordingly we find set of values of a unknown variable.

The algebra of operations over predicates

However if we want not only to learn already existing intellectual systems, but also to create new ones, the computer should not only have the ideas (relations), but also possess ability to operate with them, to set them in motion, to receive new ideas (actions over relations).

The algebra of predicates by itself cannot operate, the way of the decision of the logic equations is necessary. Only then we can obtain the information from a declarative way of representation of knowledge (the book will not train us to receive knowledge, it is necessary to apply energy). To construct the brainlike computer, the algebra of predicates is only necessary but to put it in action, *the algebra of operations over predicates* in which language it would be possible to write down such operations as formulas is necessary. Any algebra defined on the carrier M is called *algebra of predicate operations* over set of predicates M .

Roles which are carried out with algebra of predicates and algebra of operations over predicates appear precisely shared. Language of algebra of predicates is declarative, language of algebra of operations over predicates is procedural.

Logic net model

Similarly to how the person has not only ideas, but also carries out actions over them, both algebra of predicates and algebra of predicate operations are present in logic net. The multiple-valued external world demanding algebra of predicates is before us, but inside us binary logic operates that corresponds to algebra of predicate operations. Each logic net represents connection of poles and branches only, i.e. is described using pair $< X, P >$, where X – finite nonempty set of object variable (vertices), P – collection of two-place predicates (arcs), defined on set X (see Fig. 1).

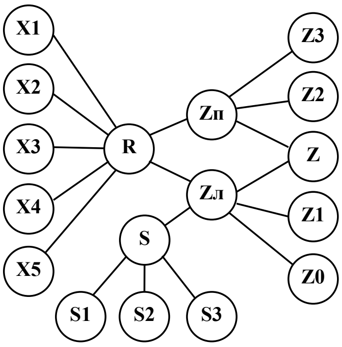

Fig. 1. Logic net of Russian language adjectives analysis

In any logic net each pole is characterized by the object variable and its definition area only, and each branch – by the two-place predicate which depends on the variables specified at the ends of this branch. Depending on complexity of a task solved, the network contains some number of poles. Accordingly, the more difficult is the task, the more poles there are in a logic network. [7] The description of object variables occurs in network nodes in language of algebra of predicates. When there is a processing of information in branches of a logic network, here algebra of predicate operations starts acting.

For hardware implementation of set x: $x \in X, x \subseteq A$, characteristic vector $W = (W_1, ..., W_j, ..., W_n)$ is used where component W_j is defined as

$$W_j = \begin{cases} 1 \leftarrow a^j \in x \\ 0 \leftarrow a^j \notin x \end{cases} \quad (4)$$

a^j – object j of set A with capacity n. Physically characteristic vector is represented as n-bit register.

For implementation of linear logic transformations between sets x and y in arcs function F is used and inverse functions F^{-1} which are defined as

$$\forall i \to W_i^y = F(W^x) = \underset{j}{\vee}[W_j^x \wedge P(a_j, a_i)] \quad (5)$$

$$\forall j \to W_j^x = F^{-1}(W^y) = \underset{i}{\vee}[W_i^y \wedge P(a_j, a_i)] \quad (6)$$

where W_i^y , W_j^x – components i and j of characteristic vectors of sets y and x; a_j, a_i – objects corresponding to respective components.

Simulation is performed in the following way: 1) in the first half-cycle linear logic transformations in the arcs of network $W^x = F(W^y), W^y = F^{-1}(W^x)$ are performed; 2) in the second half-cycle conjunction $W(t+1) = W \wedge W(t)$ of input and current vertex vectors is performed. Simulation stops when there are no differences at each vertex in the two adjacent cycles $[W(t)=W(t-1)]\rightarrow\overline{rdy}$. Schematic implementation of vertex is shown at Fig. 2. Here register RG1 stores current n-bit vector while register RG2 stores vector in t-1 cycle.

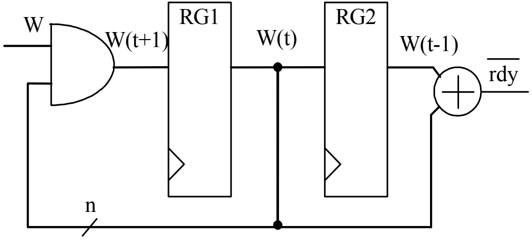

Fig.2. Schematic vertex implementation

Design process model

Design process model is show at Fig. 3. Input description is relations described using predicate algebra. Input description is transformed during compilation into internal data structures – logic net. At the synthesis stage logic net is transformed (as shown above) into Boolean equations system, suitable for hardware implementation. Output model is described using VHDL in the form of RTL netlist.

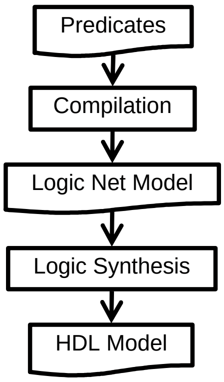

Fig. 3. Design process model

Fig. 4 shows using of logic net, which extends PC intellectual capabilities. Interaction is performed in the following way: logic net processor has high-level interface to bus through PCI controller. From the other side, device driver has application programmer interface (LogicNetAPI) to user applications, such as text processors, translation systems, speech or text recognition software.

Experimental results

For efficiency analysis, logic net simulation software was developed and evaluated on Intel Pentium IV 2.4 GHz, 256MB DDR RAM. Average simulation performance on average is 4000 words per second. It is necessary to point out, that simulation speed depends on set operations implementation features [8].

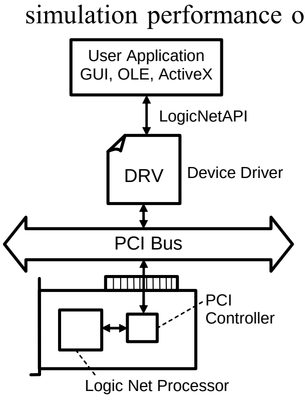

Fig. 4. Logic net and applications interaction

Logic net hardware implementation characteristics to different Xilinx PLD devices [9] are shown in the table below.

One word simulation iteration in FPGA takes from three to seven clock cycles (including initial data loading and ready signal generation). In the worst case (f=43 MHz) simulation speed is $43\cdot10^6/7\approx6\cdot10^6$ words per second, that in $6\cdot10^6/4000\approx1500$ times faster than software simulation.

Hardware implementation characteristics

	Slices, %	Luts, %	I/OBs, %	Freq, MHz
xc2s100-6-fg256	24	22	68	47
xcv50-6-fg256	37	34	68	45
xcv100-6-fg256	24	22	68	43
xcv200-6-fg256	12	11	68	41
xc2v250-6-fg256	18	17	71	63
xc2v500-6-fg256	9	8	71	65
xc2v8000-5-ff1152	<1	<1	14	43
xc2vp2-5-fg256	20	18	87	71
xc2vp4-5-fg256	9	8	87	57
xc2vp7-5-fg456	5	5	49	58

The logical and biological networks

By comparison of logic net against basic types of neurostructures invincible similarity of a structure of technical and biological designs is found out.

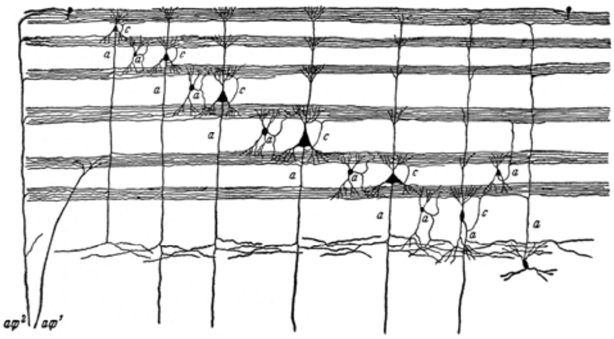

Fig. 5 and Fig. 6 demonstrate this fact evidently.

Fig. 5. Structure of connections of nervous cells

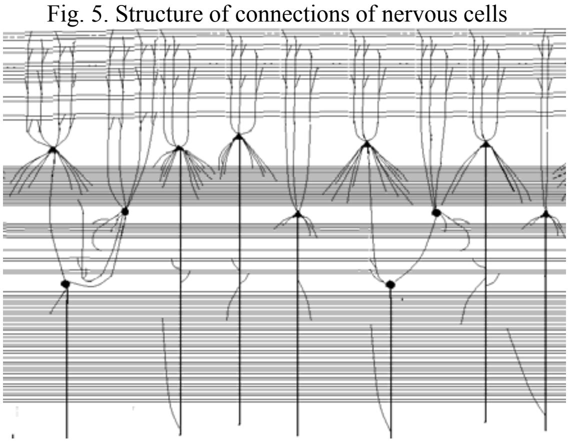

Fig. 6. Circuit realization of a logic net for Russian language adjectives

How is it possible to explain it? The matter is that the brain is neither more nor less than simulation device. It has a model of an external world with which help the brain appears capable to predict a course of events in it and, due to this the human being can survive in this world and even to subordinate nature to itself. This model can be neither more nor less than a system of relations because only by relations it is possible to describe every possible objects and processes. Therefore nature has been compelled to make the same, that people have made for the same purpose, namely to create a network for solution of equations of algebra of predicates. Nature made this network from albuminous nervous cells and fibres but not from electronic elements. Basing on this similarity, it is possible to determine functions of various types of neural structures and to describe principles of functioning of a brain in exact mathematical and technical terms.

We are far from understanding perception of objects, even from trying to state hypotheses in this field. People working in the field of artificial intellect cannot create the machine which would compete a brain in performance of such special tasks as reading the hand-written text, driving the car or recognizing faces. They have shown that solving any of these problems is connected to huge theoretical difficulties. But the problem is not in overcoming these difficulties – brain in fact copes with them obviously; the question is that methods employed cannot be simple: in language of an artificial intelligence speak it means, that tasks are nontrivial. However, it is important to have even several examples which the person would understand well: our ability to solve even a small part of processes connected with a life, demonstrates that the full understanding is essentially possible and that we do not require to appeal to mystical vital forces or to the soul [2].

Conclusions

So, the model of any mechanism of language and thinking is described with the help of system of relations, each of which is expressed by some formula of algebra of predicates. Each such a formula can be represented by some circuit which can be easily realized by existing means of radio electronics. As a result the network which has been named logic network appears. Without dependence from complexity of a task, the brainlike computer solves it for a very short time.

However, the more difficult the task is, the greater number of its computing elements is involved in operation. Limiting complexity of a task which the brainlike computer can solve is determined by the number of gates in its processor. At present the processor of the brainlike computer (it is represented by one or system of the chips connected to each other in which the logic net is realized) is traind with the help of a usual computer. Training is reduced to formation of the circuit of connections of computing elements in the chip, ordered by model realized in the brainlike computer of this or that mechanism of language and thinking. When the circuit is generated, it is possible to start the brainlike computer in to operation.

It is assumed, that in the future the brainlike computer will get ability to self-training and evolution. Its self-development will be limited to the general number of the gates available in its processor. At present personal computer is used for input of the initial data in the brainlike computer and a obtaining result from it, however, in the future the brainlike computer will be equipped with its own such devices.

This paper presents underlying principles of construction the brainlike parallel computer that surpass present serial computers on productivity by 10^{11} times, and a human brain – in 10^{6} times. New approach to making artificial intelligence: the human intelligence is considered as some material embodiment of the mechanism of logic. Works on algebraization of logic have been carried out.

This paper offers logic net model (processor of brainlike computer) of Russian adjectives oriented at hardware implementation. Design process model and logic net transformation procedures are offered. Performance analysis of model implementation into FPGA in comparison with software is performed. Speed-up gain is obtained due to the parallel data processing. Software implementation is deprived of such advantage, as long as operations in the arcs of network are performed sequentially in von Neumann architectures. Performance gain will be higher for higher parallelism model.

References:

1. *Bondarenko M.* Scientific Investigations and Studying Process in Kharkov National University of Radioelectronics. East-West Design & Test International Conference. Alushta, Crimea, Ukraine, September 17-21, 2003.

2. *Vinner N.* Cybernetics. M.: Sov. radio, 1968, p. 326.

3. *Newell A., Shaw I.C., Simon H.A.* Empirical Explorations with the Logic Theory Machine. Proceedings of the Western Joint Computer Conference. 1957, p. 218-239.

4. *Hjubel's D.* The Eye, the Brain, the Sight. M.: Mir, 1990, p. 239.

5. *Shabanov-Kushnarenko Yu.P.* Theory of intelligence. Kh: Visha shkola, 1984, 1986, 1987, p. 440.

6. *Bondarenko M., Dudar Z., Efimova I., Leshinskiy V., Shabanov-Kushnarenko S.* About Brainlike Computer. Radioelektronika i informatika. 2004. № 2.

7. *Shabanov-Kushnarenko Yu.P., Hahanov V.I., Protsay N.T., Vechirskaja I.D., Leshinskiy V., Ivanilov A.A., Obrizan V.I.* Logic Net – Technology of Natural Language Modeling. IT – in science and education. Kharkov, KHNURE, 20-22 march 2005.

8. *Zaychenko S.A., Parfentiy A.N., Kamenuka E.A., Ktiaman H.* Set Operation Speed-up of Fault Simulation. Proceedings of the 2nd East-West Design and Test Workshop 2004. Alushta, Ukraine. 23-26 September. P.231-237.

9. *The* Programmable Logic Data Book. Xilinx, Inc. 2004.

Camera-ready was prepared in Kharkov National University of Radio Electronics
by Dr. Svetlana Chumachenko and Volodymyr Obrizan
Lenin ave, 14, KNURE, Kharkov, 61166, Ukraine

Approved for publication: 05.09.2005. Format 60×841/8.
Relative printer's sheets: 33,4. Order: without cash transfer. Circulation: 100 copies.
Published by SPD FL Stepanov V.V.
Ukraine, 61168, Kharkov, Ak. Pavlova st., 311