Simulation-based verification of implementations of incompletely specified Boolean functions

Liudmila Cheremisinova

United Institute of Informatics
Problems of NAS of Belarus
Minsk, Belarus
e-mail: cld@newman.bas-net.by

ABSTRACT

The problem under discussion is to check whether a given system of incompletely specified Boolean functions is implemented by a logical description with functional indeterminacy that is represented by a system of connected blocks each of which is specified by a system of completely or incompletely specified Boolean functions. Simulation-based verification methods are considered which simulate the multi-block structure on the domain of the system of Boolean functions. The results of investigation of verification methods based on simulation of combinational circuits are given for the case of input stimuli represented by ternary vectors.

Keywords

Computer-aided design, verification, simulation.

1. INTRODUCTION

It is known currently, verification takes more than 70% efforts spent in automated electronic design [1]. So, verification becomes more and more important aspect of the design flow due to the importance of design correctness with growing VLSI design in complexity. The objective of verification is to ensure that implemented and specified behaviors are the same. In a typical scenario, there are two structurally similar circuit implementations of the same device obtained in the process of the design, and the problem is to prove their functional equivalence.

Simulation has been the main verification tool in computer-aided design systems for microelectronic devices until recently due to the simplicity of its realization and the variability of its implementation time. This approach implies the input of binary signals into the circuit being simulated, their promotion through the circuit, which corresponds to the activation of its outputs; and the examination of the results of the simulation with expected values. However, in practice, the descriptions being compared using simulation may only be analyzed partially, i.e., on a certain subset of inputs, and the completeness of the verification may not be ensured by simulation in the general case.

In recent decades, formal verification methods that aim to prove the functional identity of projects in a formal way have been developed rapidly as an alternative to simulation-based verification. Methods of verification based on its reduction to a problem of the satisfiability checking of a conjunctive normal form (CNF) [1, 2], which makes it possible to ensure the completeness of the verification, in contrast to simulation methods, are the most advanced. However, the dimensions of the CNF generated in the process of solving real-world problems often exceed the capabilities of the available SAT solvers, and software tools of formal verification are at the moment unable to to completely eliminate the necessity of using simulation-based verification tools, which provide more flexible adjustment to the dimensions of the problems being solved and de facto remain the main testing tool in design organizations.

In the paper, the verification task is examined for the more general case, when the desired functionality of the system under design is incompletely specified. Such a case usually occurs on early stages of designing when assignments to primary inputs of designed device exist which will never arise during a normal mode of the device usage. Thus, when hardware implementing this device, its outputs in response of these inputs may be arbitrary specified. In this case, verification methodologies may consider only possible input scenarios to the design under verification and verify that every possible output signal of the implemented behavior has its intended value.

We consider the verification problem for the case, when the desired incompletely specified functionality is given in the form of a system of incompletely specified Boolean functions (ISFs) and the compared functional description represents a multi-block structure with blocks specified by systems of completely or incompletely specified functions. A special case of such a multi-block structure is a combinational network or an ISF system. There is one-to-one correspondence between the arguments and functions of both compared descriptions. ISFs are specified on intervals (cubes) of values of Boolean input variables, and the intervals are large enough.

2. THE SUGGESTED APPROACH TO VERIFICATION

An ISF system $F(\boldsymbol{x}) = \{ f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x}) \}$ (where $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ is a vector) is represented as a mapping of n-dimensional Boolean space B^n of vectors $\boldsymbol{x}_i$ into m-dimensional space $\{0,1,-\}^m$ of vectors $\boldsymbol{f}_j$, where the symbol "–" denotes don't-care condition. An ISF is specified by off-set U_f^0, on-set U_f^1 and dc-set U_f^{dc} as subsets of B^n ($U_f^1 \cup U_f^0 \cup U_f^{dc} = B^n$). Let us specify a system $F(\boldsymbol{x})$ as a set I_F of multiple-output cubes $(\boldsymbol{u}, \boldsymbol{t})$ each of which is a pair of ternary vectors $\boldsymbol{u}$ and $\boldsymbol{t}$ (or conjunctions) of sizes n and m. Here $\boldsymbol{u}$ is a cube in B^n or a set of minterms $\boldsymbol{b}_i$ (elements of B^n), and $\boldsymbol{t}$ is a ternary vector of values of functions for the cube $\boldsymbol{u}$.

A system $F(\boldsymbol{x})$ of ISFs given by the set I_F of multiple-output cubes $(\boldsymbol{u}_i, \boldsymbol{t}_i)$ can be represented in matrix form by a pair of ternary matrices $\boldsymbol{U}$ and $\boldsymbol{T}$ (Fig. 1,a) or a pair of Boolean $\boldsymbol{B}$ and ternary $\boldsymbol{T}$ matrices (Fig. 1,b) of the same cardinalities. For example, ISF system (Fig. 1,a) represented in matrix

form by ternary matrices $\boldsymbol{U}$, $\boldsymbol{T}$ is specified by the set I_F = {($\bar{x}_1 x_2 \bar{x}_3$, $\bar{f}_1$), ($\bar{x}_1 \bar{x}_2 \bar{x}_3 x_4 x_5$, $f_2 f_3$), ...}.

$\boldsymbol{U}$	$\boldsymbol{T}$
$x_1x_2x_3x_4x_5$	$f_1f_2f_3$
0 1 0 – –	0 – –
0 0 0 1 1	– 1 1
1 1 – 0 0	1 – 0
1 1 1 – 1	1 0 –
0 – 1 0 1	0 – 0
1 0 0 1 1	1 1 –
1 – 1 1 0	– 0 0
1 0 0 0 –	1 1 –

a)

$\boldsymbol{B}$	$\boldsymbol{T}$
$x_1x_2x_3x_4x_5$	$f_1f_2f_3$
0 1 0 1 0	0 – –
0 0 0 1 1	– – 1
1 1 0 0 0	0 1 0
1 1 1 1 1	– 1 –
0 1 0 0 0	0 – 0
1 0 0 1 1	1 0 1
0 0 0 1 1	1 0 1
1 1 1 1 0	0 1 –
0 1 0 0 1	0 – 0
1 1 1 0 1	– 1 –
1 0 0 0 1	1 – 0

b)

Figure 1. Two forms of ISF system representation

In the paper, the task under consideration is as follows. Given an ISF system in a matrix form and a structure of the connected blocks. The task is to check whether the second form of functional representation implements the first one. We consider two cases: 1) the structure has no indeterminacy and each its block is represented by the system of conjunctive normal forms (Fig. 2); 2) the structure has indeterminacy and each its block is represented by ISF system (Fig. 3).

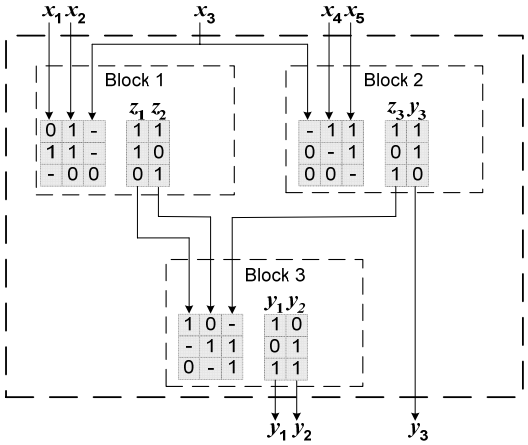

Figure 2. Three-block structure

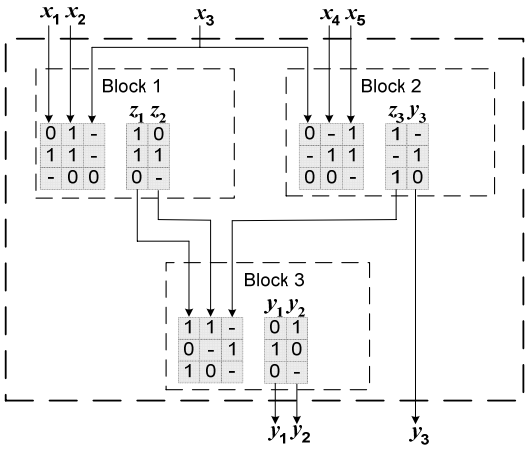

Figure 3. Three-block structure with indeterminacy

In the first case, the structure can be easily transformed into a multi-level combinational network which consists of NOT, AND and OR gates. Before simulation we fulfill levelization of the network in such a manner that before evaluating a gate, all its fan-ins would have been evaluated.

Logic simulation is the most widely used technique for verifying the correctness of digital integrated circuits in industry because of its scalability and predictable run-time behavior. This technique is based on stimulating inputs of the circuit with binary signal values that propagate in the circuit leading to a corresponding activation of the outputs, whose values must be consistent with the expected ones.

We investigate three approaches to solve the considered task: 1) binary simulation of a combinational circuit; 2) ternary simulation of a combinational circuit; 3) simulation of a multi-block structure with indeterminacy. The proposed verification methods are based on parallel simulation of the given multi-block structure (with or without indeterminacy) on the input patterns specified by the set I_F of multiple-output minterms or cubes of the compared ISF system. The multi-block structure is simulated under all possible patterns (corresponding to the elements of the domain of the system $F(\boldsymbol{x})$) simultaneously. The simulation methods are based on fast computations over long binary and/or ternary vectors.

3. BINARY SIMULATION BASED VERIFICATION

The binary simulation supposes that the verified system is specified by a set I_F of multiple-output minterms ($\boldsymbol{b}_i, \boldsymbol{t}_i$) and includes the following steps:

1) the sequential input of variable values corresponding to $\boldsymbol{b}_i$, ($\boldsymbol{b}_i, \boldsymbol{t}_i$) $\in I_F$, into the circuit;
2) the computation of the signal values at the outputs of the circuit's gates and finally at the circuit's outputs;
3) the comparison of the circuit's responses $y_j(\boldsymbol{b}_i)$ with values $\boldsymbol{t}_i$ of functions of the system F.

The ISF system F specified by the set I_F of multiple-output intervals is implementable by a circuit if the condition of covering the Boolean vector $\boldsymbol{y}(\boldsymbol{b}_i)$ by a ternary vector $\boldsymbol{t}_i$ is fulfilled for each ($\boldsymbol{b}_i, \boldsymbol{t}_i$) $\in I_F$.

The parallel simulation [3] of a multiple output logical circuit is carried out simultaneously on all the binary minterms from I_F, i.e. a state of each primary input and node of the circuit is represented by a Boolean vector of the size $|I_F|$. So, at the end of simulation we have m vectors $\boldsymbol{y}_j$ (each of the size $|I_F|$) and the only thing remaining is the orthogonality checking of all the pairs of vectors: $\boldsymbol{y}_j$ and the ternary vector $\boldsymbol{t}^j$ corresponding to the j-th column of the matrix $\boldsymbol{T}$. The implementability of the system by the combinational circuit takes place if all the pairs of vectors are not orthogonal. Otherwise, the element responsible for the violation of the implementability condition may be identified by reverse tracing of the logical circuit.

4. TERNARY SIMULATION BASED VERIFICATION

In general case the initial ISF system is specified on intervals, i.e. it is represented by a pair of ternary matrices $\boldsymbol{U}$ and $\boldsymbol{T}$ and the simulation based verification can be carried out by one of the ways: 1) by transforming the pair of ternary matrices $\boldsymbol{U}$ and $\boldsymbol{T}$ into the pair of Boolean $\boldsymbol{B}$ and ternary $\boldsymbol{T}$ matrices to have only minterms in the first matrix; 2) by solving the task directly using the interval representation.
The first way allows a binary simulation of the network under test. But when the transition to the representation of the system F on the minterms occurs, the multiple-output cube ($\boldsymbol{u}_i, \boldsymbol{t}_i$) $\in I_F$ generates 2^{n-k} multiple-output minterms ($\boldsymbol{b}_{ij}$, $\boldsymbol{t}_i$),

where k is the rank of the interval $\boldsymbol{u}_i$, and $\boldsymbol{b}_{ij}$ is the minterm of variable values covered by the interval $\boldsymbol{u}_i$. When multiple-output cubes are "expanded" and the obtained multiple-output minterms with equal output parts are grouped together, each set of the multiple-output minterms ($\boldsymbol{b}$, $\boldsymbol{t}_i$) (with the same input part $\boldsymbol{b}$) is substituted by one multiple-output minterm ($\boldsymbol{b}$, $\boldsymbol{t}$), where $\boldsymbol{t}$ is covered by each of the output parts $\boldsymbol{t}_i$ of these minterms.

Thus, the ternary matrix $\boldsymbol{U}$ is transformed into the Boolean matrix $\boldsymbol{B}$, which assigns minterms of values of input variables of the system F, and the matrix $\boldsymbol{T}$ is transformed into the matrix $\boldsymbol{T}_b$, which assigns values of ISF on these minterms. For example, the domain of definition of the system $F(\boldsymbol{x})$ (Fig. 1,a) represented by eight multiple-output cubes is transformed into a domain consisting of 16 multiple-output minterms.

The second way should be used, when the number of intervals of the set I_F having ranks less than n is big and/or these ranks are much less than n. In these cases, the resulting Boolean matrix $\boldsymbol{B}$ could be great. The use of ternary simulation makes it possible to reduce the number of simulated vectors (by up to several orders of magnitude) and, therefore, the simulation time. Further we discuss this second way as it is more time and space efficient than the first one [4].

At the beginning of the simulation, the ordered set of n ternary vectors (corresponding to the columns of the matrix $\boldsymbol{U}$ and having the size $|I_F|$) are taken as network inputs. Then gates of the network are simulated in the predefined topological order. Let a gate implementing the function φ_i (z_{1i}, z_{2i}, …, z_{ki}) be simulated. For each its argument z_{ji} a ternary vector $\boldsymbol{z}_{ji}$ corresponds to and the vectors $\boldsymbol{z}_{ji}$ have been computed already. So, the simulation of the gate is reduced to performing the logic operation φ_i over ternary vectors $\boldsymbol{z}_{1i}$, $\boldsymbol{z}_{2i}$, …, $\boldsymbol{z}_{ki}$ in the bitwise style. The result of the simulation is a new ternary vector $\boldsymbol{z}_i$ of the same size. The definition of basic operations over ternary variables is given bellow for all combinations of values of two ternary variables:

$\boldsymbol{a}$:	0	0	0	–	–	–	1	1	1
$\boldsymbol{b}$:	0	–	1	0	–	1	0	–	1
$\bar{\boldsymbol{a}}$:	1	1	1	–	–	–	0	0	0
$\boldsymbol{a} \wedge \boldsymbol{b}$:	0	0	0	0	–	–	0	–	1
$\boldsymbol{a} \vee \boldsymbol{b}$:	0	–	1	–	–	1	1	1	1

As soon as the last gate of the network has been simulated the following pairs of vectors are compared: the ternary vector $\boldsymbol{f}^j$ (i = 1, 2,…, m) of values of the function $f_j \in F$ (the vector $\boldsymbol{f}^j$ corresponds to the j-th column of the matrix $\boldsymbol{T}$) and the ternary vector $\boldsymbol{y}_j$ corresponding to the j-th primary output of the network. The following three cases are possible:

1. Vectors $\boldsymbol{f}^j$ and $\boldsymbol{y}_j$ are orthogonal in some component. Hence, the network does not implement the function f_j.
2. The vector $\boldsymbol{f}^j$ covers the vector $\boldsymbol{y}_j$, i.e. values of all definite components of $\boldsymbol{f}^j$ are the same as the values of the corresponding components of $\boldsymbol{y}_j$. The network implements the function f_j.
3. The value of some i-th component of the vector $\boldsymbol{y}_j$ is don't care, while the value of the corresponding component of the vector $\boldsymbol{f}^j$ (corresponding to the interval $\boldsymbol{u}_i$ of the matrix $\boldsymbol{U}$) is equal 1 or 0. In this case, there exists no unambiguous answer whether the network implements the function f_j or not.

In the last case, an additional analysis is needed to detect the reason of distinction between the values. The simplest way is to simulate the network once more on all minterms of the interval $\boldsymbol{u}_i \in \boldsymbol{U}$. Another way is to analyze controversial interval $\boldsymbol{u}_j$ using SAT based verification methods [1, 5].

For instance, in Fig. 4 the results of the parallel ternary simulation of the three-block structure (Fig. 2) on cubes of the ISF system (Fig. 1,a) are presented. When the simulation is finished, the three pairs of vectors setting the values of the functions of the ISF system and the functions implemented at the structure's outputs must be compared:

$\boldsymbol{y}_1$: 0 1 1 1 – 1 0 1 $\quad$ $\boldsymbol{y}_2$: – 1 0 0 0 1 0 1 $\quad$ $\boldsymbol{y}_3$: – 1 0 – 0 1 0 –
$\boldsymbol{f}^1$: 0 – 1 1 **0** 1 – 1 $\quad$ $\boldsymbol{f}^2$: – 1 – 0 – 1 0 1 $\quad$ $\boldsymbol{f}^3$: – 1 0 – 0 – 0 –

$x_1x_2x_3x_4x_5$	$z_1z_2z_3$	$y_1y_2y_3$
0 1 0 – –	1 1 –	0 – –
0 0 0 1 1	0 1 1	1 1 1
1 1 – 0 0	1 0 –	1 0 0
1 1 1 – 1	1 0 –	1 0 –
0 – 1 0 1	1 – 0	– 0 0
1 0 0 1 1	0 1 1	1 1 1
1 – 1 1 0	1 0 0	1 0 0
1 0 0 0 –	0 1 1	1 1 –

Figure 4. Results of parallel simulation of the three-block structure (Fig. 2)

The aforementioned condition 2 is fulfilled for both pairs except the 5-th component of $\boldsymbol{f}^1$ (it is shown in $\boldsymbol{f}^1$ in bold face) for which the case 3 takes place. But by splitting the interval $\boldsymbol{u}_5$ = 0 – 1 0 1 into two minterms and simulating the structure on them we find out that y_1(0 0 1 0 1) = y_1(0 1 1 0 1) = 0 = t_5^1, i.e. y_1 implements f_1.

5. SIMULATION OF A MULTI-BLOCK STRUCTURE WITH INDETERMINACY

The second considered case when each block of the multi-block structure realizes an ISF system (Fig. 3) is more complex and it is considered [6] for the case when the verified ISF system is specified on the domain of minterms only (specified by a Boolean matrix $\boldsymbol{B}$ instead of the ternary one).

ISF f can be represented by a pair of disjunctive normal forms collecting conjunctions on which the function f takes values 1 and 0, correspondingly. So, each block can be considered as a two-level multi-output combinational network. The first is formed by multi-input AND gates, implementing conjunctions, and in the second level for each function y_i^k of the k-th block, a pair of multi-input OR gates is used: one of them to implement y_i^k and the other – to implement $\bar{y}_i^k$. Inputs of the OR gate implementing the function y_i^k (and $\bar{y}_i^k$) are fed upon outputs of those AND gates which implement conjunctions on which the function y_i^k takes value 1 (correspondingly, $y_i^k = 0$).

To get an ISF a pseudo-element is introduced – two input UNITE gate. Such a gate joins signals from two OR gates of the second level for y_i^k and $\bar{y}_i^k$. Keep in mind, that a pair of completely specified Boolean functions $y_i^1(\boldsymbol{x})$ и $y_i^0(\boldsymbol{x})$ defines the ISF $y_i(\boldsymbol{x})$ which takes the value 1 on a minterm $\boldsymbol{b}_j$, if $y_i^1(\boldsymbol{b}_j) = 1$, the value 0, if $y_i^0(\boldsymbol{b}_j) = 1$, the value is don't care, if $y_i^1(\boldsymbol{b}_j) = y_i^0(\boldsymbol{b}_j) = 0$, and $y_i^1(\boldsymbol{b}_j) = y_i^0(\boldsymbol{b}_j) = 1$ does not take place for consistent assignment of ISF, the UNITE function $f(x_0,x_1)$ could be specified as follows:

x_0:	0	0	1	1
x_1:	0	1	0	1
$f(x_0,x_1)$:	–	1	0	*

Just as in the case 1, the multi-block structure (Fig. 3) is transformed into a multy-level network consisting of invertors, AND, OR and UNITE gates. The network is levelized and then simulated on the set of Boolean vectors corresponding to the columns of the matrix $\boldsymbol{U}$. After finishing the simulation, the value y_i of every i-th primary output of the network is compared with the value of the corresponding i-th component t_j^i of the row $\boldsymbol{t}_j$ of the matrix $\boldsymbol{T}$ for each j. The following cases are possible.

1. t_j^i = "–" or $y_i(\boldsymbol{b}_j) = t_j^i$ for all j. In this case, the structure implements the function f_i.
2. $t_j^i = \sigma$ and $y_i(\boldsymbol{b}_j) = \bar{\sigma}$ or $y_i(\boldsymbol{b}_j)$ = "–" ($\sigma \in \{1, 0\}$) for some j. In this case, the structure does not implement the function f_i.

For example, the result of simulation of the structure shown in Fig. 3 using the set of 5 binary vectors (corresponding to the columns of the matrix $\boldsymbol{B}$ of ISF system (Fig. 1,b)) allows to conclude that it implements the tested ISF system because only the case 1 has a place:

$\boldsymbol{y}_1$: 0 1 0 0 0 1 1 0 0 0 1 $\quad$ $\boldsymbol{y}_2$: – 0 1 1 – 0 0 1 – 1 0;
$\boldsymbol{f}^1$: 0 – 0 – 0 1 1 0 0 – 1 $\quad$ $\boldsymbol{f}^2$: – – 1 1 – 0 0 1 – 1 –
$\boldsymbol{y}_3$: – 1 0 1 0 1 1 – 0 – 0
$\boldsymbol{f}^3$: – 1 0 – 0 1 1 – 0 – 0

6. INVESTIGATION RESULTS OF COMPARING SIMULATION-BASED METHODS OF VERIFICATION

As it can be seen previously, ternary simulation is a powerful tool for verification because it allows simulate the verified network on intervals of Boolean space (bundles of minterms). But using the ternary simulation for some multiple-output cubes for which case 3 takes place, it is impossible to clearly answer whether the circuit implements some function or not. And so, the additional analysis is needed, for instance, binary simulation on minterms from this multiple-output cube.

To estimate the viability of using ternary simulation for verification, it is important to answer the question: to what extent is ternary simulation efficient in terms of reducing the space of minterms for which binary simulation is unavoidable.

In order to evaluate the efficiency of ternary simulation, investigations have been carried out on a stream of 94 examples [7], each of which is an ISF system and a network implementing it. To carry out the experiments, pseudorandom ISF systems with predetermined parameters (the number of variables, functions, intervals, and the average number of don't-cares in them) were generated.

The synthesis of combinational circuit was carried out using two design procedures that begin with ISF system minimization and its decomposition into multilevel network of primitive gates, and end with transition to network consisting of elements from the CMOS library. The difference between the design procedures is that the first procedure [8] fulfils the decomposition into a multilevel network of two input NANDs, while the second one [9] fulfils the algebraic decomposition into a multilevel network of multiple-input ANDs and ORs.

During the verification based on ternary simulation, the average amount k% of intervals (out of their total number) from the ISF system definition domain was registered for which ternary simulation failed to provide a result. The experiments have shown that the amount of intervals verified using ternary simulation depends more strongly on the synthesis technique of the circuit than on the parameters of the initial ISF system. The first technique turned out to be the most "inconvenient" synthesis technique for verification based on ternary simulation. This may be caused by the fact that the initial intervals of the Boolean space are split strongly in the process of multilevel network synthesis. With this fact discovered, the authors focused on this "inconvenient" technique in experiments (73 of 94 examples) in order to examine the efficiency of ternary simulation in worst cases.

Nevertheless, the experimental results have shown that ternary simulation makes it possible to verify a significant part of all intervals. On the average, the ternary simulation made it possible to verify the implementability of 76% of the examined intervals of ISF systems generated and implemented by circuits using both techniques, and 69% of the examined intervals of ISF systems generated and implemented by circuits using the first "inconvenient" design procedure.

The obtained results make it possible to conclude that the use of ternary simulation for the verification of logical description defined on a set of intervals of a Boolean space may significantly reduce the part of the space to be verified using more time consuming techniques (such as binary simulation or SAT-based verification).

REFERENCES

[1] A. Wiemann, *Standardized functional Verification*, Springer, San Carlos, CA USA, 2008.
[2] A. Kuehlmann, A.J. Cornelis van Eijk, "Combinational and Sequential Equivalence Checking", in Logic synthesis and Verification (Ed. S. Hassoun, T. Sasao, R.K. Brayton), Kluwer Academic Publishers, pp. 343–372, 2002.
[3] A. Zakrevskij, Yu. Pottosin, L. Cheremisinova, *Design of Logical Control Devices*, TUT Press, Tallinn, 2009.
[4] L. Cheremisinova, D. Novikov, "Simulation-based approach to verification of logical descriptions with functional indeterminacy", *Information Theories & Applications*, FOI ITHEA, Bulgaria, pp. 218–224, V. 15, No. 3, 2008.
[5] L.D. Cheremisinova, D. Ya. Novikov, "Formal Verification with Functional Indeterminacy on the Basis of Satisfiability Testing of the Conjunctive Normal Form", *Automatic Control and Computer Sciences*, Allerton Press, Inc., pp. 1–10, Vol. 44, No. 1, 2010.
[6] L.D. Cheremisinova, D.Ya. Novikov, "Verification of Multi-block structures with functional indeterminacy", *Proceedings of the NAS of Belarus, physical-technical series*, Minsk, pp. 98–105, No 2, 2009 (in Russian).
[7] D.Ya. Novikov, L.D. Cheremisinova, "Investigation of Simulation-Based Verification Methods for Descriptions with Functional Indeterminacy", *Automatic Control and Computer Sciences*, Allerton Press Inc., pp. 191–199, Vol. 46, No. 5, 2012.
[8] E.M. Sentovich E.M., K.J. Singh, L. Lavagno, et al, "SIS: A System for Sequential Circuit Synthesis", 1992, http://www. eecs.berkeley.edu/Pubs/TechRpts/1992/ERL-92-41.pdf.
[9] P.N. Bibilo, L.D. Cheremisinova, S.N. Kardash, et al, "Low-Power Synthesis of Logical CMOS Circuits", *Proc.of the 5th All-Russia Science&Technology Conf. "Problems of Advanced Micro- and Nanoelectronic Systems Development"*, IPPM RAN, Moscow, pp. 73–78, 2012 (in Russian).