Liudmila CHEREMISINOVA
THE UNITED INSTITUTE OF INFORMATICS PROBLEMS OF NAS OF BELARUS,
ul. Surganova 6, 220012 Minsk

Verification of logical descriptions with functional indeterminacy

D.Sc. Liudmila CHEREMISINOVA

DSc. Degree from the United Institute of Informatics Problems of NAS of Belarus (2000). Principal researcher of the Laboratory of Logical Design of the Institute (2002). Scientific results: about 250 publications, ten books. Research interests: VLSI and discrete devices design automation, logical control of concurrent processes, formal verification, topological optimization, low-power synthesis.

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 blocks is specified by a system of completely or incompletely specified Boolean functions. Simulation based and SAT based verification methods is considered. The first method simulates the structure specified by the second description on the domain of the first description. The second method formulates the verification problem as checking satisfiability of a conjunctive normal form. The results of computer investigation of the proposed methods are given.

Keywords: design automation, formal verification, simulation.

Weryfikacja specyfikacji logicznych z indeterminizmem funkcjonalnym

Streszczenie

W artykule omówiono problem sprawdzania, czy dany układ częściowo określonych funkcji Boole'owskich jest realizowany przez specyfikację logiczną z indeterminizmem funkcjonalnym. Ta specyfikacja jest przedstawiona jako system połączonych bloków, z których każdy odpowiada układowi całkiem albo częściowo określonych funkcji Boole'owskich. Rozpatrzono metodę symulacyjną i metodę bazującą na analizie spełnialności funkcji. Pierwsza z tych metod symuluje strukturę, opisaną przez drugą specyfikację, w dziedzinie pierwszej specyfikacji. Druga metoda sprowadza problem weryfikacji do problemu spełnialności funkcji w postaci iloczynu sum. Przedstawiono wyniki komputerowych badań skuteczności zaproponowanych metod.

Słowa kluczowe: automatyzacja projektowania, formalna weryfikacja, symulacja.

1. Introduction

Verification becomes more and more important aspect of the design flow due to the importance of design correctness with growing VLSI design in complexity. Currently, verification takes more than 70% efforts spent in automated electronic design. 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 design, and the problem is to prove their functional equivalence.

In contrast to that in the paper, the verification task is examined for the case, when 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 must consider only possible input scenarios to the design under verification and verify that every possible output signal of the implemented behavior has its intended (described in initial specification) value.

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

ISFs are specified on intervals (cubes) of values of Boolean input variables, and the intervals are large enough. Such a statement of the verification problem occurs in logical design of combinational part of logical devices when an indeterminacy of an initial large ISF system is gradually decreased from step to step of design process. After each successive step the behaviors of two compared descriptions must be equivalent only on those sets of values of Boolean input variables (elements of Boolean space), where Boolean functions of the first system are specified.

2. Preliminaries

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 into m-dimensional space $\{0,1,-\}^m$, 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^{ds} as subsets of B^n ($U_f^1 \cup U_f^0 \cup U_f^{ds} = B^n$). 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 that further are called its input and output parts. The input part $\boldsymbol{u}$ is a cube in B^n or a set of minterms (elements of B^n), the output part $\boldsymbol{t}$ is a ternary vector of values of functions for the cube $\boldsymbol{u}$.

The system $F(X)$ 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) or a pair of Boolean $\boldsymbol{B}$ and ternary $\boldsymbol{T}$ matrices (Fig. 2) of the same cardinalities. For example, ISF system $F(\boldsymbol{x}) = (\boldsymbol{U}, \boldsymbol{T})$ (Fig. 1) is specified by the set $I_F = \{(\bar{x}_1 x_2 \bar{x}_3, \bar{f}_1), (\bar{x}_1 \bar{x}_2 \bar{x}_3 x_4 x_5, f_2 f_3), \ldots\}$ of multiple-output cubes.

We are focusing on the case in which the first description is an ISF system and the second of the compared descriptions is an implementation of the first one and is represented by some sort of multi-block structure. Further we consider two cases: 1) the structure has no indeterminacy and each its block is represented by CNF system (Fig. 1); 2) the structure has indeterminacy and each its block is represented by ISF system (Fig. 2).

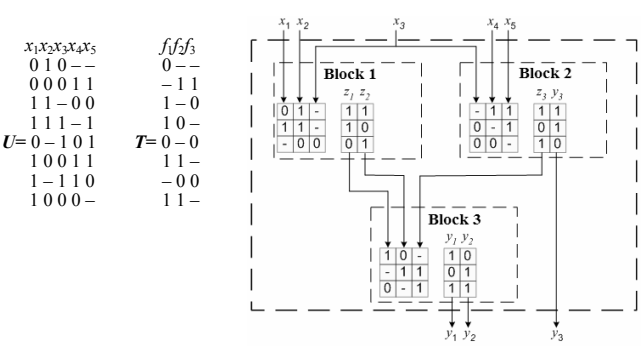


Fig. 1. An example of ISF system and a three-block structure implementing it
Rys. 1. Przykład systemu ISF i trzyblokowa struktura implementująca

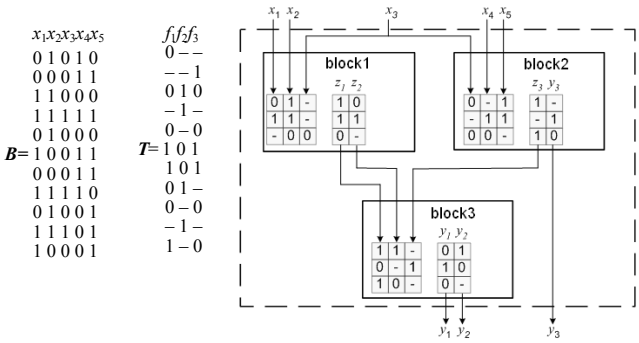


Fig. 2. An example of ISF system and a three-block structure with indeterminacy
Rys. 2. Przykład systemu ISF i trzyblokowa struktura z nieokreślonością

3. Simulation based verification: case 1

Logic simulation is the most widely used technique for ensuring the correctness of digital integrated circuits in industry because of its scalability and predictable run-time behavior. This technique is based on verifying a digital system by 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.

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 cubes of the compared ISF system. The multi-block structure is simulated under all possible inputs (corresponding to the elements of the domain of the system $F(\boldsymbol{x})$) simultaneously, i.e. a state of each primary input and node of the circuit is represented by a Boolean or a ternary vector of the size $|I_F|$. The simulation is based on the fast Boolean computations over long binary and/or ternary vectors [1, 2].

In the first considered case (Fig. 1) each block of the multi-block structure realizes a system of disjunctive normal forms. The structure can be easily transformed into a multi-level combinational network which consists of NOT, AND and OR gates. Before simulation the network gates are levelized such a manner that before a gate is evaluated, all its fan-ins would have been evaluated.

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 by one of the ways: 1) by transforming the pair of ternary matrices $\boldsymbol{U}$ and $\boldsymbol{T}$ into the a 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 Boolean simulation of the network S under test. 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 $\boldsymbol{B}$ can be great. Further we discuss this second way as it is more time and space efficient than the first one [1].

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 S are simulated in the predefined topological order. Let a gate implementing the function $\varphi_i(z_{1i}, z_{2i}, \ldots, z_{ki})$ is 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 $\boldsymbol{z}_{1i}, \boldsymbol{z}_{2i}, \ldots, \boldsymbol{z}_{ki}$ 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 $\boldsymbol{t}^i$ (i = 1, 2,…, m) corresponding to the i-th column of the matrix $\boldsymbol{T}$ (the column of values of the function $f_i \in F$) and the ternary vector $\boldsymbol{y}_p$ corresponding to the primary output y_p of the network S. The following three cases are possible:

1. Vectors $\boldsymbol{t}^i$ and $\boldsymbol{y}_i$ are orthogonal in some component. Hence, the network S does not implement the function f_i.
2. The vector $\boldsymbol{t}^i$ covers the vector $\boldsymbol{y}_i$, i.e. values of all definite components of $\boldsymbol{t}^i$ are the same as the values of the corresponding components of $\boldsymbol{y}_i$. The network implements the function f_p.
3. The value of some j-th component of the vector $\boldsymbol{y}_i$ is don't care, while the value of the corresponding component of the vector $\boldsymbol{t}^i$ (corresponding to the interval $\boldsymbol{u}_j$ of the matrix $\boldsymbol{U}$) is equal 1 or 0. In this case, there exists no unambiguous answer whether the network S implements the function f_i or does 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 S once more on all minterms of the interval $\boldsymbol{u}_j \in \boldsymbol{U}$. Or to analyze controversial intervals $\boldsymbol{u}_j$ using SAT based verification method.

For example, simulation of the structure shown in Fig. 1 gives:

f_1: 0 – 1 1 **0** 1 – 1	f_2: – 1 – 0 – 1 0 1	f_3: – 1 0 – 0 – 0 –
y_1: 0 1 1 1 – 1 – 1	y_2: – 1 0 0 0 1 0 1	y_3: – 1 0 – 0 1 0 –

The only exception is in the 5-th component of f_1 for which the case 3 takes place. By splitting the interval $\boldsymbol{u}_5$ = 0–101 into two minterms and simulating the structure on them we find out $y_1(00101) = y_1(01101) = 0 = t_5^1$, i.e. y_1 implements f_1.

4. Simulation based verification: case 2

The second considered case (Fig. 2) when each block of the multi-block structure realizes an ISF system is more complex and it is considered [2] 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 its inversion y_i^k. Inputs of the OR gate implementing function y_i^k (and 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 y_i^k. Keep in mind, that a pair of completely defined Boolean functions $y_i^1(X)$ и $y_i^0(X)$ specifies the ISF $y_i(X)$ which takes the value 1 on a minterm $\boldsymbol{b}_j$, if $y_i^1(\boldsymbol{b}_j) = 1$, the value 0, if $y_i^0(\boldsymbol{b}_j) = 1$, and the value of $y_i(X)$ is don't care, if $y_i^1(\boldsymbol{b}_j) = y_i^0(\boldsymbol{b}_j) = 0$, 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 –

It should be noted that the last combination takes no place for consistent assignment of ISFs specifying blocks.

Just as in the case 1, the multi-block structure (Fig. 2) is tranformed 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. $t_j^i = \sigma$ 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.
2. $t_j^i =$ "–" or $y_i(\boldsymbol{b}_j) = \boldsymbol{t}_j^i$ for all j. In this case, the structure implements the function f_i.

For example, the result of simulation of the structure shown in Fig. 2 allows to conclude that it implements the tested ISF system:

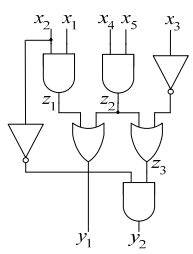
f_1: 0 – 0 – 0 1 1 0 0 – 1 f_2: – – 1 1 – 0 0 1 – 1 – f_3: – 1 0 – 0 1 1 – 0 – 0
y_1: 0 1 0 0 0 1 1 0 0 0 1 y_2: – 0 1 1 – 0 0 1 – 1 0 y_3: – 1 0 1 0 1 1 – 0 – 0

5. SAT based approach to verification

The past ten years have seen efforts in developing commercial formal verification tools that provide more general results than traditional simulation methods: it is possible to guarantee that a specific property holds for a design under all possible input stimuli. In a typical scenario, there are two structurally similar implementations of the same design, and the problem is to prove their functional equivalence [3]. In a modern combinational equivalence checking flow based on formal verification approach, both networks to be verified are transformed into a single comparing circuit. It is derived by combining the pairs of inputs with the same names and feeding the pairs of outputs with the same names into EXOR gates, which are ORed to produce the single output of the comparing circuit. There is constant 0 on the output if and only if the two original circuits are equivalent.

A conjunctive normal form (CNF) represents a Boolean function as conjunction of one or more clauses, each being in its turn a disjunction of literals. The SAT problem is concerned with finding a truth assignment of literals which simultaneously satisfies each of CNF clauses. If such an assignment exists the CNF is referred to as satisfiable, and the assignment is known as a satisfying assignment.

To test whether the circuit output be 1 or 0, the transformation is applied to it for producing its conventional CNF. Once the overall problem is formulated in CNF, a SAT solver can be used to solve it [3, 4]. A circuit-to-CNF conversion uses as many variables as there are primary inputs and gates in the circuit: for output of each gate its own internal Boolean variable is introduced. And a local CNF is associated with each gate, the CNF captures the consistent assignments between its inputs and output. Then local CNFs are joined in the overall network CNF $C(S)$ by the conjunction operation (Fig. 3).



$x_1x_2x_3x_4x_5z_1z_2z_3y_1y_2$

1 – – – – 0 – – – – 1
– 1 – – – 0 – – – – 2
0 0 – – – 1 – – – – 3
– – – 1 – – 0 – – – 4
– – – – 1 – 0 – – – 5
– – – 0 0 – 1 – – – 6
– – 0 – – – 1 0 – – 7
– – 1 – – – – 1 – – 8
– – – – – – 0 1 – – 9
– – – – – 1 1 – 0 – 10
– – – – – 0 – – 1 – 11
– – – – – – 0 – 1 – 12
– 0 – – – – – – – 0 13
– – – – – – – 1 – 0 14
– 1 – – – – – 0 – 1 15

Fig. 3. An example of combinational circuit and its conventional CNF
Rys. 3. Przykład układu kombinacyjnego i jego konwencjonalna spójna forma normalna C

The derivation of CNF for a gate representing a local function $y = f(z_1, z_2, \ldots, z_k)$ is based on defining a new Boolean function $\varphi(y,f) = y \sim f$ [3], that is true in the only case when both functions y and f assume the same value. Here are the conventional CNF representations of NOT, n-input AND and OR functions:

$(z \vee y)(\bar{z} \vee \bar{y})$;
$(z_1 \vee \bar{y})(z_2 \vee \bar{y}) \ldots (z_n \vee \bar{y})(\bar{z}_1 \vee \bar{z}_2 \vee \ldots \vee \bar{z}_n \vee y)$;
$(\bar{z}_1 \vee y)(\bar{z}_2 \vee y) \ldots (\bar{z}_n \vee y)(z_1 \vee z_2 \vee \ldots \vee z_n \vee \bar{y})$.

Note, that the above traditional approach which constructs comparing circuit can not be applied for the case considered in the paper because of both given forms of functional representation can be specified incompletely.

6. SAT based verification: case 1

A problem under discussion is to verify if a given network implements the ISF system. It is true if it takes place for each multiple-output cube. In terms of network CNF this condition could be reformulated as follows [5]. For every multiple-output cube $(\boldsymbol{u}_i, \boldsymbol{t}_i) \in I_F$ a partial value assignment satisfying the conjunction $\boldsymbol{u}_i \boldsymbol{t}_i$ should satisfy the network CNF.

In other words, a network implements ISF system $F(\boldsymbol{x})$, if for every multiple-output cube $(\boldsymbol{u}_i, \boldsymbol{t}_i) \in I_F$ a partial value assignment satisfying the conjunction $\boldsymbol{u}_i\ \bar{\boldsymbol{t}}_i$ (i.e. contradicting to $\boldsymbol{u}_i, \boldsymbol{t}_i$) is unsatisfying assignment for the network CNF. If $\boldsymbol{u}_i = x_1^i x_2^i \ldots x_{ni}^i$ and $\boldsymbol{t}_i = f_1^i f_2^i \ldots f_{mi}^i$ then the CNF P_i specifying the contradiction of the multiple-output cube $(\boldsymbol{u}_i, \boldsymbol{t}_i)$, called as the cube-prohibitive CNF, consists of the following $n_i + 1$ clauses:

$$P_i(\boldsymbol{x},\boldsymbol{f}) = x_1^i x_2^i \ldots x_{mi}^i (\bar{f}_1^i \vee \bar{f}_2^i \vee \ldots \vee \bar{f}_{mi}^i).$$

For example, the prohibitive CNF for the multiple-output cube $s_6 = (x_1 x_2, f_1 \bar{f}_2)$ from $F(\boldsymbol{x}) = (\boldsymbol{U}, \boldsymbol{T})$ (Fig. 4) has three clauses: $P_6(\boldsymbol{x},\boldsymbol{f}) = (x_1)(x_2)(\bar{f}_1 \vee f_2)$.

Appending clauses of P_i to the network CNF $C(S)$ results in CNF $C(P_i) = C(S) \wedge P_i$. It is not difficult to prove that CNF $C(P_i)$ is satisfiable if the network does not implement the i-th multiple-output cube. As to the whole ISF system it is not implemented by the network if at least one its multiple-output cube is not implemented by the network, i.e. if the following CNF is satisfiable:

$$C = C(S) \wedge P(F) = C \wedge (P_1 \vee P_2 \vee \ldots \vee P_l), \qquad (2).$$

where $P(F)$ is the ISF system prohibitive CNF.

$x_1x_2x_3x_4x_5$

– – 1 1 1
– 0 0 0 –
– 0 – 1 1
0 1 – 1 0
– 0 1 0 –
1 1 – – –

$\boldsymbol{U} =$

f_1f_2

1 – 1
0 1 2
– 1 3
0 0 4
– 0 5
1 0 6

$\boldsymbol{T} =$

$x_1x_2x_3x_4x_5f_1f_2w_1w_2w_3w_4w_5w_6$

– – 1 – – – – 1 – – – – – 1
– – – 1 – – – 1 – – – – – 2
– – – – 1 – – 1 – – – – – 3
– – – – – 0 – 1 – – – – – 4
– 0 – – – – – – 1 – – – – 5
– – 0 – – – – – 1 – – – – 6
– – – 0 – – – – 1 – – – – 7
– – – – – 1 0 – 1 – – – – 8
…
– – – – – – – 0 0 0 0 0 0 25

$P^K =$

Fig. 4. An example of ISF system $(\boldsymbol{U}, \boldsymbol{T})$ and its encoded prohibitive CNF P^K
Rys. 4. Przykład systemu ISF $(\boldsymbol{U}, \boldsymbol{T})$ i jego zakodowana wersja zakazana

To apply any SAT-solver to check whether for the CNF C a satisfying assignment exists it is necessary to convert the formula $P(F)$ to a CNF form. Theoretically this could be done always, but it is NP-hard problem. We are interested in a method of construction of ISF system prohibitive CNF $P(F)$ having linear complexity. Next the method is proposed that is based on encoding multiple-output cubes and their prohibitive CNFs using coding variables $w_i \in \boldsymbol{w}$. After encoding, prohibitive CNFs $P_i(\boldsymbol{x},\boldsymbol{f})$ are transformed into encoded prohibitive CNFs $P_i^K(\boldsymbol{x},\boldsymbol{f},\boldsymbol{w})$ and the formula (2) into

$$C^{\kappa} = C \wedge (P_1^k \wedge P_2^k \wedge \ldots \wedge P_l^k) \wedge Q(\boldsymbol{w}), \quad (3)$$

where $Q(\boldsymbol{w})$ provides that the CNF C^{κ} will be satisfiable if at least one CNF $P_i \in P(F)$ is satisfiable. From now on Q is called as alternative CNF.

When transforming the formula (2) into the CNF form (3) each cube-prohibitive CNF P_i is encoded by a code in the form of a disjunction $d_i = w_{i1}^{\sigma i1} \vee w_{i2}^{\sigma i2} \vee \ldots \vee w_{ir}^{\sigma ir}$ ($\sigma_{ir} \in \{0,1\}$, $w_{ir}^1 = w_{ir}$ and $w_{ir}^0 = \bar{w}_{ir}$, and $w_{ij} \in \boldsymbol{w}$:

$$P_i^k(\boldsymbol{x}, \boldsymbol{f}, \boldsymbol{w}) = (x_1^i \vee d_i) \ldots (x_{ni}^i \vee d_i)(\bar{f}_1^i \vee \ldots \vee \bar{f}_{mi}^i \vee d_i). \quad (4)$$

To formulate the conditions the alternative CNF $Q(\boldsymbol{w})$ in (3) must satisfy for the chosen cube-prohibitive CNF encoding, let us denote by f_Q and f_{di} the functions represented by $Q(\boldsymbol{w})$ and $d_i(\boldsymbol{w})$ and by U_Q^1 and U_{di}^1 – their on-sets.

Assertion [6]. Any alternative CNF $Q(\boldsymbol{w})$ for a given encoding of cube-prohibitive CNFs must satisfy the following conditions:

1) $(\bigwedge_i f_{di}) \wedge f_Q = 0$ or $(\bigcap_i M_{di}^1) \cap M_Q^1 = \varnothing$;

2) $(\bigwedge_{i \neq j} f_{di}) \wedge f_Q \neq 0$ or $(\bigcap_{i \neq j} M_{di}^1) \cap M_Q^1 \neq \varnothing$ for all j.

The first condition ensures the CNF $P(\boldsymbol{x}, \boldsymbol{f}, \boldsymbol{w}) = (P_1^k \wedge P_2^k \wedge \ldots \wedge P_l^k) \wedge Q$ be unsatisfiable when the circuit implements the analyzed ISF system, i.e. when all cube-prohibitive CNFs $P_i(\boldsymbol{x}, \boldsymbol{f})$ are unsatisfiable. The second condition ensures the CNF $P(\boldsymbol{x}, \boldsymbol{f}, \boldsymbol{w})$ be satisfiable when the circuit do not implement the analyzed ISF system, i.e. there exists at least one multiple-output cube, for example j-th one, that is not realized by it. Thus, a variable assignment can be found satisfying the cube-prohibitive CNF $P_j(\boldsymbol{x}, \boldsymbol{f})$ (and $P_j^k(\boldsymbol{x}, \boldsymbol{f}, \boldsymbol{w})$ too). Fulfillment of the second condition guaranties that there exists at least one assignment of coding variables that ensures satisfiability of $Q(\boldsymbol{w})$ and all cube prohibitive CNFs P_i^k except the j-th one (that is satisfiable by the assumption).

Two basic methods of encoding multiple-output cubes (satisfying the above Assertion) have been investigated: encoding by codes of unit [5] and logarithmic length [7]. The first method supposes to introduce as many coding variables w_i as there exist multiple-output cubes in the ISF system specification I_F. The second method introduces the minimal number of coding variables that is $r = \lceil log_2 I_F \rceil$. The method of encoding by codes of logarithmic length allows to reduce substantially the number of coding variables as compared with the method of encoding by unary codes, but codes (and CNF clauses) are dense enough.

Further we focus upon increasing efficiency of verification process using unary encoding of multiple-output cubes. Using it $P_i^k(\boldsymbol{x}, \boldsymbol{f}, \boldsymbol{w})$ (4) changes for the following encoded form:

$$P_i^{\kappa} = (x_1^i \vee w_i)(x_2^i \vee w_i) \ldots (x_{ni}^i \vee w_i)(\bar{f}_1^i \vee \ldots \vee \bar{f}_{mi}^i \vee w_i),$$

and the alternative CNF Q satisfying the above Assertion can be:

$$Q = \bar{w}_1 \vee \bar{w}_2 \vee \cdots \vee \bar{w}_l.$$

For example, the fragment of the prohibitive CNF for the ISF system in the matrix form is shown in Fig. 4. If we combine the circuit conventional CNF (Fig. 3) and the prohibitive CNF (identifying y_1, y_2 with f_1, f_2) and then carry out the satisfiability test of the resulting CNF, we may make sure that it is nonsatisfiable. Therefore, the circuit (Fig. 3) implements the ISF system (Fig. 4).

Three verification methods are proposed [8]: based on successive, simultaneous and group testing multiple-output cubes from I_F. The first method formulates as many SAT problems as the number of cubes are there, the second formulates verification task as the only SAT problem (using coding the cubes as shown above), the third divides the overall set I_F of multiple-output cubes into groups and formulates as many SAT problems as the number of groups are there. The group method is more effective because it allows 1) to achieve trade-offs between expenses on forming data for SAT-solver and SAT-solver performance; and thereby 2) to reduce overall verification time [8].

7. SAT based verification: case 2

Here we consider the verification problem for the case, when both compared descriptions are incompletely specified. For example, we have a multi-block structure S with indeterminacy such as one in Fig. 2 and an ISF system $F(\boldsymbol{x})$ such as one in Fig. 4. Just as in the previous section we formulate the verification problem as verifying whether CNF $C = C(S) \wedge P$ is satisfiable. Here P is the ISF system $F(\boldsymbol{x})$ prohibitive CNF, $C(S)$ is some sort of the conventional CNF but only for a multi-block structure with indeterminacy or for a ISF system (that can be considered instead of one-block structure). To distinguish it from the conventional CNF let call it further as the permissible CNF. The CNF describes the set of admissible combinations of signals on all the nodes of structure blocks; i.e., each set of values satisfying the CNF is admissible for this structure. The permissible CNF $C(S)$ is the conjunction of permissible CNFs $C(B_i)$ of its blocks or permissible CNFs $C(F_i)$ their ISF systems.

Three methods of construction of a permissible CNF for an ISF system are proposed: one based on the paraphrased representation of ISFs [9], and two based on the application of implicative conditions: implication [10] and implication with condition coding [11] methods. Here we dwell on the implication method.

Assertion [10]. The permissible CNF $C(G)$ of an ISF system $G(\boldsymbol{x})$ defined by a set of its multiple-output cubes $s_i = (\boldsymbol{u}_i, \boldsymbol{t}_i)$ $(i = 1, 2, \ldots, r)$ is generated by the formula:

$$(\boldsymbol{u}_1 \rightarrow \boldsymbol{t}_1) \wedge (\boldsymbol{u}_2 \rightarrow \boldsymbol{t}_2) \wedge \ldots \wedge (\boldsymbol{u}_r \rightarrow \boldsymbol{t}_r).$$

The permissible CNF for a multiple-output cube $s_i^g = (\boldsymbol{u}_i, \boldsymbol{t}_i)$ with $\boldsymbol{u}_i = x_1^i x_2^i \ldots x_{ni}^i$ and $\boldsymbol{t}_i^g = y_1^i y_2^i \ldots y_{mi}^i$ consists of as many clauses as the size of the term $\boldsymbol{t}_i$ is:

$$C_i = (\boldsymbol{u}_i \rightarrow \boldsymbol{t}_i) = \bar{\boldsymbol{u}}_i \vee \boldsymbol{t}_i = \bar{x}_1^i \vee \bar{x}_2^i \vee \ldots \vee \bar{x}_{ni}^i \vee (y_1^i y_2^i \ldots y_{mi}^i) =$$
$$= (\bar{x}_1^i \vee \bar{x}_2^i \vee \ldots \vee \bar{x}_{ni}^i \vee y_1^i) \wedge \ldots \wedge (\bar{x}_1^i \vee \bar{x}_2^i \vee \ldots \vee \bar{x}_{ni}^i \vee y_{mi}^i).$$

For example, the permissible CNF for the first block of the structure (Fig. 2) consists of five clauses: $(x_1 \vee \bar{x}_2 \vee z_1) \wedge (x_1 \vee \bar{x}_2 \vee \bar{z}_2)(x_1 \vee x_2 \vee z_1)(x_1 \vee x_2 \vee z_2)(\bar{x}_2 \vee \bar{x}_3 \vee \bar{z}_1)$.

The key idea of the proposed method of checking whether an ISF system $F(\boldsymbol{x})$ is implemented by a multi-block structure S with indeterminacy is as follows. We obtain the prohibitive CNF $P(F)$ and the permissible CNF $C(S)$.

Assertion. A multi-block structure S with indeterminacy implements an ISF system $F(\boldsymbol{x})$ if the CNF $P(F) \wedge C(S)$ is unsatisfiable.

One can make sure after constructing $P(F) \wedge C(S)$ for the ISF system in Fig. 4 and three-block structure in Fig.2 that there is no satisfying assignment for the CNF. Thus the structure implements by the ISF system.

8. Experimental results

All the mentioned verification methods have been implemented on C++ programming language. Then the programs were investigated on the sets of pseudo-random pairs of descriptions: ISF system and multi-block structure implementing it (with or without indeterminacy). MiniSat solver [4] has been used in our experiments. The goal of experiments was 1) to compare the competitive methods solving the same task on the same set of examples; 2) to investigate experimentally domains of preferable usage of the proposed methods; 3) to find out the most effective value of group size for group testing verification methods. The experiments have shown that:

1) simulation based verification methods have 60 times greater speed on average than SAT based methods solving the same task;
2) the group size about 200 gives good enough results: group methods gain stably in efficiency compared with the methods of successive and simultaneous testing of multiple-output cubes, the win gain is about 35% over the method of simultaneous testing;
3) substantial reduction of variables when using logarithmic encoding of multiple-output cubes did not bring about substantial speedup of the solution of verification problem;
4) despite the fact that the implication method is simpler than that of implication with condition coding and gives shorter CNFs, it has smaller speed.

9. References

[1] Cheremisinova L., Novikov D.: International Journal "Information Theories & Applications". FOI ITHEA, Bulgaria, V. 15, No. 3, 2008.
[2] Cheremisinova L.D., Novikov D.Ya.: Proceedings of the NAS of Belarus, physical-technical series, Minsk, No 2, 2009 (in Russian).
[3] Kunz W., Marques Silva J., Malik S.: Logic synthesis and Verification (Ed. S.Hassoun, T.Sasao and R.K.Brayton). Kluwer Academic Publishers, 2002.
[4] The MiniSat Page / http://minisat.se/MiniSat.html
[5] Cheremisinova L., Novikov D.: Proceedings of 8th Intern. Workchop on Boolean problems, Freiberg (Sachsen), Sept. 18–19, 2008.
[6] Cheremisinova L., Novikov D.: Proc. of IEEE East-West Design and Test Symposium (EWDTS'09), Moscow, Russia, Sept. 18–21, 2009.
[7] Cheremisinova L., Novikov D.: International book series "Information science and computing". FOI ITHEA, Bulgaria, No 15, 2009.
[8] Cheremisinova L.D., Novikov D.Ya.: Automatic Control and Computer Sciences. Allerton Press, Inc., Vol. 44, No. 1, 2010.
[9] Novikov D.Ya., Cheremisinova L.D.: Informatika. The United Institute of Informatics Problems of NAS of Belarus, No. 3, 2010 (in Russian).
[10][10] Cheremisinova L., Novikov D.: Proc. of 9th Int.. Workshop on Boolean problems, Freiberg (Sachsen), Sept. 16–17, 2010.
[11] Cheremisinova L.D., Novikov D.Ya.: Automatic Control and Computer Sciences, Allerton Press, Inc., Vol. 45, No. 4, 2011.

otrzymano / received: 15.03.2012
przyjęto do druku / accepted: 01.05.2012 *artykuł recenzowany / revised paper*

INFORMACJE

Newsletter PAK

Wydawnictwo PAK wysyła drogą e-mailową do osób zainteresowanych Newsletter PAK, w którym są zamieszczane:
- spis treści aktualnego numeru miesięcznika PAK,
- kalendarz imprez branżowych,
- ważniejsze informacje o działalności Wydawnictwa PAK.

Newsletter jest wysyłany co miesiąc do osób, które w jakikolwiek sposób współpracują z Wydawnictwem PAK (autorzy prac opublikowanych w miesięczniku PAK, recenzenci, członkowie Rady Programowej, osoby które zgłosiły chęć otrzymywania Newslettera).

Celem inicjatywy jest umocnienie w środowisku pozycji miesięcznika PAK jako ważnego i aktualnego źródła informacji naukowo-technicznej.

Do newslettera można zapisać się za pośrednictwem:
- strony internetowej: www.pak.info.pl, po dodaniu swojego adresu mailowego do subskrypcji,
- adresu mailowego: wydawnictwo@pak.info.pl, wysyłając swoje zgłoszenie.

Otrzymywanie Newslettera nie powoduje żadnych zobowiązań ze strony adresatów. W każdej chwili można zrezygnować z otrzymywania Newslettera.

Tadeusz SKUBIS
Redaktor naczelny Wydawnictwa PAK

Nowy dział „Niepewność wyników pomiarów" na stronie internetowej Wydawnictwa PAK

Uprzejmie informuję, że na stronie internetowej Wydawnictwa PAK (WWW.pak.info.pl) został utworzony dział „Niepewność wyników pomiarów". Na p.o. redaktora działu został powołany dr inż. Paweł Fotowicz.

Dr P. Fotowicz jest ekspertem w zakresie problematyki niepewności, autorem szeregu wartościowych publikacji w czasopismach krajowych i zagranicznych. Prezentował swoje prace na licznych konferencjach i warsztatach szkoleniowych.

W dziale „ Niepewność wyników pomiarów", obok dostępu do aktualnych wybranych opracowań dotyczących niepewności jest możliwość zadawania „Pytań do eksperta". Pytania powinny być konkretne i szczegółowo sprecyzowane.

Pytania i odpowiedzi o istotnym znaczeniu dla szerszego grona metrologów będą archiwizowane i dostępne dla użytkowników strony internetowej Wydawnictwa PAK. Zapraszam do odwiedzania działu „Niepewność wyników pomiarów" i do udziału w jego rozwoju.

Tadeusz SKUBIS
Redaktor naczelny Wydawnictwa PAK