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

Streszczenie

W artykule omówiono problem sprawdzania, czy dany układ częściowo określonych funkcji Boole'owskich jest realizowany przez specyfikację logiczną z indeterminizmem funkcjonalnym, która 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, weryfikacja, symulacja.

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 and SAT based verification methods are considered. The first methods simulate the structure specified by the second description on the domain of the first description. The second methods formulate 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.

1. Introduction

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.

We consider the verification problem for the case, when desired functionality is given in the form of a system of incompletely specified Boolean functions (ISFs) and the compared functional description is given in the form of a multi-block structure that consists of connected blocks each of them represents a system of completely or incompletely specified Boolean functions. Such a statement of the verification problem occurs in logical design of combinational part of logical devices when an indeterminacy of an initial ISF system is gradually decreased from step to step of design process. Two approaches to solve the verification task are investigated: simulation and formal verification by reducing to SAT problem.

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. 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 $\boldsymbol{u}$.

2. Simulation based verification

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.

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 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 a node of the circuit is represented by a Boolean or a ternary vector of the size $|I_F|$. The simulation is based on fast Boolean computations over long binary and/or ternary vectors [1, 2].

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. To implement an ISF a pseudo-element is introduced – two input UNITE gate that joins signals from two input OR gate implementing functions y_i^k and $\bar{y}_i^k$. The UNITE function could be specified as follows: UNITE(1,0)=1, UNITE(0,1)=0, UNITE(0,0)=UNITE(1,1)= "–".

In the case when all blocks of the structure represent completely specified functions the structure can be viewed as a combinational network consisting of NOT, AND and OR gates. Two input UNITE gates appear in the network if some blocks of the structure realize ISFs. 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 is more time and space efficient than the first one [1], so it is used but only in the case of purely combinational network (without UNITE gates).

At the beginning of the simulation, the ordered set of n ternary vectors (having the size $|I_F|$) are taken as network inputs. The simulation of any gate is reduced to performing the logic operation over ternary vectors $\boldsymbol{z}_{1i}, \boldsymbol{z}_{2i}, \ldots, \boldsymbol{z}_{ki}$ (that are the gate inputs) in the bitwise style [1].

As soon as the last gate of the network has been simulated, the analysis of simulation results is made: the network S does or does not implement the ISF system. In some cases there exists no unambiguous answer and then the additional analysis is needed. The simplest way is to simulate the network S once more on all minterms of the controversial intervals or to analyze the structure behavior on these intervals using SAT based verification method.

3. SAT based approach to verification

The past ten years have seen efforts in developing commercial formal verification tools (by reducing to SAT) that provide more general results than traditional simulation methods. 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 both networks to be verified are transformed into a single comparing circuit such that there is the constant 0 on its output iff two original circuits are equivalent. To test whether the comparing circuit output be 1 or 0, its *conventional* conjunctive normal form (CNF) is produced applying the circuit-to-CNF conversion [3]. Two circuits under comparison are equivalent iff the comparing circuit conventional CNF is unsatisfiable (there is no satisfying assignment).

The traditional approach can not be applied for the considered case as at least one of compared functional descriptions can be incompletely specified. To reduce the verification problem to SAT we construct two CNFs $P(F)$ and $C(S)$. CNF $P(F)$ describes all assignments contradictive to the first form (ISF system) and is called *prohibitive* CNF of the ISF system. CNF $C(S)$ describes all possible assignments for the second form (multi-block structure), and it is called *conventional* CNF [3] in the case of the structure without indeterminacy (combinational circuit) or otherwise it is called *permissible* CNF that is some sort of the conventional CNF for a structure with indeterminacy.

Assertion. The multi-block structure implements ISF system if and only if CNF $P(F) \wedge C(S)$ is unsatisfiable [4, 6, 7].

A network implements ISF system $F(\boldsymbol{x})$, iff for every multiple-output cube $(\boldsymbol{u}_i, \boldsymbol{t}_i) \in I_F$ a value assignment satisfying the conjunction $\boldsymbol{u}_i\ \overline{\boldsymbol{t}_i}$ (i.e. contradicting to $\boldsymbol{u}_i, \boldsymbol{t}_i$) is unsatisfying assignment for the network CNF. If $\boldsymbol{u}_i = x_1^{\,i} x_2^{\,i} \dots x_{ni}^{\,i}$ and $\boldsymbol{t}_i = f_1^{\,i} f_2^{\,i} \dots f_{mi}^{\,i}$ then the cube-prohibitive CNF P_i consists of the $n_i + 1$ clauses: $P_i(\boldsymbol{x}, \boldsymbol{f}) = x_1^{\,i} x_2^{\,i} \dots x_{ni}^{\,i} (\overline{f}_1^{\,i} \vee \overline{f}_2^{\,i} \vee \dots \vee \overline{f}_{mi}^{\,i})$. The ISF system *prohibitive* CNF $P(F)$ is functionally equvivalent to the function $P_1 \vee P_2 \vee \dots \vee P_l$. The formula could be directly converted into a CNF form, but that is NP-hard problem. The method of linear complexity is proposed that is based on coding multiple-output cubes and their prohibitive CNFs using Boolean variables $w_i \in \boldsymbol{w}$ and codes in the form of disjunctions $d_i = w_{i1}^{\sigma i1} \vee w_{i2}^{\sigma i2} \vee \dots \vee w_{ir}^{\sigma ir}$ ($\sigma_{ir} \in \{0,1\}$, $w_{ir}^{\,1} = w_{ir}$ and $w_{ir}^{\,0} = \overline{w}_{ir}$). After encoding, we get the ISF system prohibitive CNF $P(\boldsymbol{x}, \boldsymbol{f}, \boldsymbol{w}) = (P_1^{\,k} \wedge P_2^{\,k} \wedge \dots \wedge P_l^{\,k}) \wedge Q(\boldsymbol{w})$, where $P_i^{\,k}(\boldsymbol{x}, \boldsymbol{f}, \boldsymbol{w}) = (x_1^{\,i} \vee d_i) \dots (x_{ni}^{\,i} \vee d_i)(\overline{f}_1^{\,i} \vee \dots \vee \overline{f}_{mi}^{\,i} \vee d_i)$ and the CNF $Q(\boldsymbol{w})$ called as alternative CNF provides that the CNF $P(\boldsymbol{x}, \boldsymbol{f}, \boldsymbol{w})$ will be satisfiable iff at least one CNF $P_i \in P(F)$ is satisfiable.

To formulate the conditions the alternative CNF $Q(\boldsymbol{w})$ must satisfy for the chosen cube-prohibitive CNF encoding, let 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 $P(\boldsymbol{x}, \boldsymbol{f}, \boldsymbol{w}) \wedge C(S)$ 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. 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 [4] and logarithmic length [5]. 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. Usage of unary encoding generates the following expressions for $P_i^{\,k}(\boldsymbol{x}, \boldsymbol{f}, \boldsymbol{w})$ and $Q(\boldsymbol{w})$ satisfying the above Assertion:

$$P_i^{\,k}(\boldsymbol{x}, \boldsymbol{f}, \boldsymbol{w}) = (x_1^{\,i} \vee w_i)(x_2^{\,i} \vee w_i) \dots (x_{ni}^{\,i} \vee w_i)(\overline{f}_1^{\,i} \vee \dots \vee \overline{f}_{mi}^{\,i} \vee w_i),$$
$$Q(\boldsymbol{w}) = \overline{w}_1 \vee \overline{w}_2 \vee \dots \vee \overline{w}_l.$$

Three verification methods are proposed [6]: 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 [6].

In the case when the multi-block structure S has indeterminacy we formulate the verification problem as testing whether CNF $P(F) \wedge C(S)$ is unsatisfiable [7] where $C(S)$ is the permissible CNF. The CNF $C(S)$ describes the set of admissible combinations of signals on all the nodes of the structure S blocks. The *permissible* CNF $C(S)$ is the conjunction of permissible CNFs $C(B_i)$ of its blocks or permissible CNFs $C(F_i)$ of 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, and two based on the application of implicative conditions: implication and implication with condition coding methods [7]. The simplest of them, the implication method, is based on permissible CNF definition. The permissible CNF $C(F)$ of the ISF $F(\boldsymbol{x})$ system specified by the set of its multiple-output cubes $s_i = (\boldsymbol{u}_i, \boldsymbol{t}_i)$ $(i = 1, 2, \dots, r)$ is generated by the formula:

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

Having in view that $\boldsymbol{u}_i = x_1^{\,i} x_2^{\,i} \dots x_{ni}^{\,i}$, $\boldsymbol{t}_i^{\,g} = y_1^{\,i} y_2^{\,i} \dots y_{mi}^{\,i}$ and $(\boldsymbol{u}_i \rightarrow \boldsymbol{t}_i) = \overline{\boldsymbol{u}}_i \vee \boldsymbol{t}_i = (\overline{x}_1^{\,i} \vee \overline{x}_2^{\,i} \vee \dots \vee \overline{x}_{ni}^{\,i} \vee y_1^{\,i}) \wedge \dots \wedge (\overline{x}_1^{\,i} \vee \overline{x}_2^{\,i} \vee \dots \vee \overline{x}_{ni}^{\,i} \vee y_{mi}^{\,i})$ we can easily obtain permissible CNF $C(F_i)$ and $C(S)$.

4. 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). 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.

5. 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] W. Kunz, J. Marques-Silva, S. Malik.: Logic synthesis and Verification (Ed. S.Hassoun, T.Sasao and R.K.Brayton). Kluwer Academic Publishers, 2002.

[4] Cheremisinova L., Novikov D.: Proceedings of 8th Intern. Workchop on Boolean problems, Freiberg (Sachsen), Sept. 18–19, 2008.

[5] Cheremisinova L., Novikov D.: International book series “Information science and computing”. FOI ITHEA, Bulgaria, No 15, 2009.

[6] Cheremisinova L.D., Novikov D.Ya.: Automatic Control and Computer Sciences. Allerton Press, Inc., Vol. 44, No. 1, 2010.

[7] Cheremisinova L.D., Novikov D.Ya.: Automatic Control and Computer Sciences, Allerton Press, Inc., Vol. 45, No. 4, 2011.

otrzymano / received: 00.00.0000
przyjęto do druku / accepted: 00.00.0000 *artykuł recenzowany*