SAT based approaches to verification of logical descriptions with functional indeterminacy

Cheremisinova L.D.
The laboratory of logical design
The United Institute of Informatics Problems of National Academy of Sciences of Belarus
Minsk, Belarus
e-mail: cld@newman.bas-net.by

***Abstract*—The problem under consideration 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. SAT based verification methods are considered that formulate the verification problem as checking satisfiability of a conjunctive normal form.**

Keywords-design automation; verification; testing

I. INTRODUCTION

It is known currently, verification takes more than 70% efforts spent in automated electronic design [1]. 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.

The approach to solve the verification task is investigated, that is based on its reducing to SAT problem.

II. 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. 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 $\boldsymbol{u}$.

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 a system of conjunctive normal forms (CNF); 2) the structure has indeterminacy and each its block is represented by ISF system.

III. THE SUGGESTED 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 [1]. 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 [1]. Two circuits under comparison are equivalent iff the comparing circuit conventional CNF is unsatisfiable (there is no satisfying assignment).

The traditional approach cannot 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 [1] 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 [2, 4].

A. *SAT based verification: case 1*

A problem under discussion is to verify whether 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: for every multiple-output cube $(\boldsymbol{u}_i, \underline{\boldsymbol{t}_i}) \in I_F$ a value assignment satisfying the conjunction $\boldsymbol{u}_i\ \boldsymbol{t}_i$ (i.e. contradicting to $\boldsymbol{u}_i, \boldsymbol{t}_i$) is unsatisfying assignment for the network CNF. If $\boldsymbol{u}_i = x_1{}^i x_2{}^i \ldots x^i{}_{ni}$ and $\boldsymbol{t}_i = f_1{}^i f_2{}^i \ldots f_{mi}{}^i$ then the cube-prohibitive CNF P_i consists of the $n_i + 1$ clauses:

$$P_i(\boldsymbol{x},\boldsymbol{f}) = x_1{}^i x_2{}^i \ldots x^i{}_{ni} (\, \overline{f_1}{}^i \vee \overline{f_2}{}^i \vee \ldots \vee \overline{f}_{mi}{}^i).$$

The ISF system prohibitive CNF $P(F)$ is functionally equivalent to the function $P_1 \vee P_2 \vee \ldots \vee P_l$. 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 $d_i = w_{i1}^{\sigma i1} \underline{\vee}\, w_{i2}^{\sigma i2} \vee \ldots \vee\ w_{ir}^{\sigma ir}$ ($\sigma_{ir} \in \{0,1\}$, $w_{ir}^{1} = w_{ir}$ and $w_{ir}^{0} = \overline{w}_{ir}$). After encoding, we get the ISF system prohibitive CNF

$$P(\boldsymbol{x}, \boldsymbol{f}, \boldsymbol{w}) = (P_1^{\,k} \wedge P_2^{\,k} \wedge \ldots \wedge P_l^{\,k}) \wedge Q(\boldsymbol{w}),$$

where $P_i^{\,k}(\boldsymbol{x}, \boldsymbol{f}, \boldsymbol{w}) = (x_1^{\,i} \vee d_i) \ldots\ (x^{i}_{ni} \vee d_i)(\ \overline{f}_1^{\,i} \vee \ldots \vee\ \overline{f}_{mi}^{\,i} \vee d_i)$ 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 [3]. 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 [2] and logarithmic length [4]. 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:

$$P_i^{\,\kappa}(\boldsymbol{x}, \boldsymbol{f}, \boldsymbol{w}) =$$

$$= (x_1^{\,i} \underline{\vee}\, w_i)(\underline{x}_2^{\,i} \vee w_i) \ldots \underline{\ }(x^{i}_{ni} \vee w_i)(\ \overline{f}_1^{\,i} \vee \ldots \vee\ \overline{f}_{mi}^{\,i} \vee w_i),$$

$$Q(\boldsymbol{w}) =\ \ w_1 \vee\ \ w_2 \vee \cdots \vee\ \ w_l\,.$$

Three verification methods are proposed [5]: 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 [5].

B. SAT based verification: case 2

Here we consider the verification problem for the case, when both compared descriptions are incompletely specified, i.e. the multi-block structure S has indeterminacy. Just as in the previous section, we formulate the verification problem as verifying whether CNF $P(F) \wedge C(S)$ is satisfiable [6]. Here $C(S)$ is the permissible CNF that 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 [7], and two based on the application of implicative conditions: implication and implication with condition coding methods [6]. The simplest of them, the implication method, is based on permissible CNF definition.

Assertion [6]. The permissible CNF $C(F_i)$ of an ISF system $F_i\,(\boldsymbol{x})$ 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:

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

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

IV. Experimental results

The program implementations of the mentioned verification methods 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) 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 [5];

2) substantial reduction of variables, provided by logarithmic encoding of multiple-output cubes, did not bring about substantial speedup of verification;

3) despite the fact that the implication method is simpler, than that of implication with condition coding, and gives shorter CNFs, it has smaller speed.

[1] W. Kunz, J. Marques-Silva, S. Malik, “and ATPG: Algorithms for Boolean Decision Problems”, Logic synthesis and Verification (Ed. S.Hassoun, T.Sasao and R.K.Brayton), Kluwer Academic Publishers, 2002, pp. 309–341.

[2] L. Cheremisinova, D. Novikov, “SAT-Based Approach to Verification of Logical Descriptions with Functional Indeterminacy”, Proc. 8th Intern. Workchop on Boolean problems, Freiberg (Sachsen, Germany), Sept. 18–19, 2008, pp. 59–66.

[3] Cheremisinova L., Novikov D. “SAT-Based Group Method for Verification of Logical Descriptions with Functional Indeterminacy”, Proc. 7th IEEE East-West Design & Test Sympos (EWDTS 2009), Moscow (Russia), Sept. 18–21, 2009, pp. 31–34.

[4] L. Cheremisinova, D. Novikov, “SAT-based method of verification using logarithmic encoding”, Intern. book series “Information science and computing”. FOI ITHEA, Bulgaria, 2009, No 15, pp. 107–114.

[5] Л.Д. Черемисинова, Д.Я. Новиков, “Формальная верификация описаний с функциональной неопределенностью на основе проверки выполнимости конъюнктивной нормальной формы”, Автоматика и вычислительная техника, 2010, № 1, с. 5–16.

[6] Д.Я. Новиков, Л.Д. Черемисинова, “Анализ реализуемости описаний с функциональной неопределенностью на основе проверки выполнимости конъюнктивной нормальной формы”, Автоматика и вычислительная техника, 2011, № 4, с. 36–48.

[7] Л.Д. Черемисинова, Д.Я. Новиков, “Верификация функциональных описаний с неопределенностью на основе парафазного представления булевых функций”, Информатика, 2010, № 3(27), с. 54 – 62.