ПРОЕКТИРОВАНИЕ И ДИАГНОСТИКА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

УДК 681.5.09
DOI: 10.17223/19988605/39/11

A. Matrosova, E. Mitrofanov, T. Shah

SIMPLIFICATION OF FULLY DELAY TESTABLE COMBINATIONAL CIRCUITS AND FINDING OF PDF TEST PAIRS

Работа выполнена при финансовой поддержке Российского научного фонда (проект № 14-19-00218).

Fully delay testable circuits obtained by covering ROBDD nodes with Invert-AND-OR sub-circuits and Invert-AND-XOR sub-circuits implementing Shannon decomposition formula are considered. Algorithms of finding test pairs for robust testable PDFs and validatable non robust testable PDFs of resulted circuits have been developed. Experimental results demonstrate essential simplification of suggested circuits in contrast to fully delay testable circuits obtained by covering each ROBDD node with only Invert-AND-XOR sub-circuit.
Keywords: path delay fault (PDF); robust testable PDF; validatable non robust testable PDF; Binary Decision Diagram (BDD); design for testability.

Path delay fault (PDF) model is considered as more preferable at delay testing. In accordance with the conditions of fault manifestation, single PDFs are divided into robust testable faults and non robust testable faults. PDF is robust testable if there is a test pair fault manifestation of which does not depend on delays of other circuit paths. PDF is non robust testable if fault manifestation is possible only when all other paths of a circuit are fault-free.

PDF testing has become a very important problem along with development of nanometer technologies. It is very important to provide testability for robust PDF during circuit design. Circuits derived from ROBDDs are usually implemented using multiplexors (MUXs). Their testability is investigated under different fault models [1–4] but the approaches suggested do not provide 100% testability. In [5] simple transformation of a circuit is suggested that guarantees 100% testability for both single stuck-at fault (SAF) and PDF models. These circuits are derived from ROBDDs using multiplexors. A size of a circuit is proportional to the given ROBDD size. The major disadvantage of this approach is the use of additional input.

In [6] it is shown that a circuit constructed from BDD by covering CLBs guarantees 100% testability for robust PDFs without an additional input.

In [7] the combinational circuits constructed from ROBDDs by Invert-AND-XOR implementation of the formula $f_v = \overline{x}_i f_v^{x_i=0} \oplus x_i f_v^{x_i=1}$ corresponding to an internal node v are considered. In this formula, the operation "⊕" is implemented by XOR gate. It is revealed that each path delay fault of the resulted circuit manifests itself either as robust testable fault or as validatable non robust testable one. When applying the test pairs in the definite order, we may detect any PDF of the circuit. This means that the circuits considered guarantee 100% testability for PDFs without an additional input.

In this paper the combinational circuits constructed from ROBDDs by covering some internal nodes with Invert-AND-OR sub-circuits implementing the formula $f_v = \overline{x}_i f_v^{x_i=0} \vee x_i f_v^{x_i=1}$ and covering the rest internal nodes with Invert-AND-XOR sub-circuits implementing the formula $f_v = \overline{x}_i f_v^{x_i=0} \oplus x_i f_v^{x_i=1}$ are considered. When using this approach, it is possible to cut path lengths of the combinational circuits and cut the number of OR, AND, NOT gates if we implement XOR as a sub-circuit from these gates. It is revealed that PDFs in the resulted circuits (similar circuits are considered in [7]) manifest themselves either as robust testable or as validatable non

robust testable ones. When applying test pairs in the definite order, we may detect any PDF of the circuit. This means that the circuits suggested in this paper (as the circuits in [7]) guarantee 100% testability for PDFs without an additional input. The experimental results showed that the suggested circuits as a rule were simpler than the ones in [Ibid.].

In Section II the problem of deriving the proper combinational circuits is discussed. In Section III an algorithm of ROBDD internal node analysis is suggested. In Section IV the properties of the formula originated by the circuit are investigated. In Section V algorithms of finding test pairs on which PDF manifests itself either as robust testable or validatable non robust testable one are proposed. In Section VI the experimental results are given.

1. A combinational circuit design

It is well known that Binary Decision Diagram (BDD) is a directed acyclic graph based on Shannon decomposition in each non terminal node ***v***:

$$f_v = \overline{x_i} f_v^{x_i=0} \vee x_i f_v^{x_i=1},$$
$$f_v^{x_i=0} = f_v(x_1,...,x_i = 0,...,x_n),$$
$$f_v^{x_i=1} = f_v(x_1,...,x_i = 1,...,x_n).$$

Here f_v is the function corresponding to the node *v*, dashed edge points to $f_v^{x_i=0}$ and bold edge points to $f_v^{x_i=1}$ (Fig. 1). BDD is called ordered if variables are encountered in the same order on all paths connecting the BDD root with the terminal node. BDD is reduced if it does not contain either isomorphic sub-graphs or nodes such that $f_v^{x_i=0} = f_v^{x_i=1}$. Reduced and ordered BDD (ROBDD) is a canonical representation of Boolean function for the chosen order of variables [8].

Any path that connects the BDD root with the 1 terminal node creates the product of the Disjoint Sum of Products (DSoP) of a function *f* represented by this ROBDD. DSoP is a sum of products in which any two product cubes do not intersect.

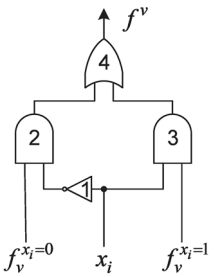

Fig. 1. Gate implementation of the formula $f_v = \overline{x_i} f_v^{x_i=0} \vee x_i f_v^{x_i=1}$

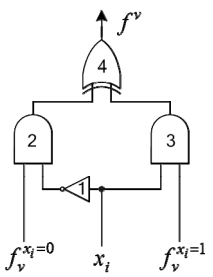

Fig. 2. Gate implementation of the formula $f_v = \overline{x_i} f_v^{x_i=0} \oplus x_i f_v^{x_i=1}$

Let $F = \{f_1,...,f_m\}$, be a system of Boolean functions describing combinational circuit behavior. Derive ROBDD using the same order of variables for each Boolean function from *F*. Join isomorphic sub-graphs in the different ROBDDs. Combine the 1 terminal nodes of the different ROBDDs into one 1 terminal node and their 0 terminal nodes into one 0 terminal node. As a result, we obtain the graph with *m* roots and two terminal nodes. This graph represents the system of *m* Boolean functions. It is called Shared ROBDD. Without loss of generality, we further consider systems with one function.

The ROBDD for one output Boolean function is shown in Fig. 1. Find the product (cube) of the DSoP for each path connecting the ROBDD root with the 1 terminal node. The DSoP of the function *f* is as follows.

$$f = x_1\overline{x}_2\overline{x}_3 \vee x_1\overline{x}_2x_3\overline{x}_4x_5 \vee x_1\overline{x}_2x_3x_4\overline{x}_5 \vee x_1x_2\overline{x}_4x_5 \vee$$
$$\vee x_1x_2x_4\overline{x}_5 \vee \overline{x}_1\overline{x}_2\overline{x}_4x_5 \vee \overline{x}_1\overline{x}_2x_4\overline{x}_5 \vee \overline{x}_1x_2\overline{x}_3\overline{x}_4x_5 \vee$$
$$\vee \overline{x}_1x_2\overline{x}_3x_4\overline{x}_5 \vee \overline{x}_1x_2x_3\overline{x}_5.$$

Eliminate from the ROBDD all edges connected with the 0 terminal node and obtain the ROBDD representing combinational circuit behavior. Call this ROBDD as a Circuit ROBDD.

Cover each node of the Circuit ROBDD with either Invert-AND-OR sub-circuit implementing the formula $f_v = \overline{x_i} f_v^{x_i=0} \vee x_i f_v^{x_i=1}$ or Invert-AND-XOR sub-circuit implementing the formula $f_v = \overline{x_i} f_v^{x_i=0} \oplus x_i f_v^{x_i=1}$. Note that both these formulae represent the same Boolean function.

The condition $f_v^{x_i=0} \neq f_v^{x_i=1}$ is satisfied for each internal node *v* of the ROBDD. This means that there exists the Boolean vector γ on which either $f_v^{x_i=1}(\gamma)=1$ and $f_v^{x_i=0}(\gamma)=0$ or $f_v^{x_i=1}(\gamma)=0$ and $f_v^{x_i=0}(\gamma)=1$.

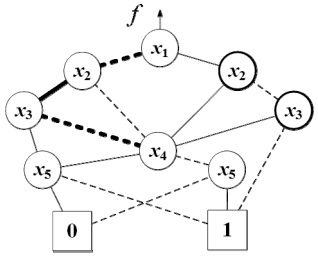

Fig. 3. ROBDD for *f*

If for the internal node *v* there exists the Boolean vector γ on which $f_v^{x_i=1}(\gamma)=1$ and $f_v^{x_i=0}(\gamma)=0$ and the Boolean vector δ on which $f_v^{x_i=1}(\delta)=0$ and $f_v^{x_i=0}(\delta)=1$ we cover this node with Invert-AND-OR sub-circuit (Fig. 2). Unfortunately sometimes only one condition pointed above is satisfied. The latter is possible when one of the functions $f_v^{x_i=0}$, $f_v^{x_i=1}$ is implicant of another one: either $f_v^{x_i=0} \leq f_v^{x_i=1}$ or $f_v^{x_i=1} \leq f_v^{x_i=0}$. In that case, we cover the corresponding internal node with Invert-AND-XOR sub-circuit (Fig. 3).

2. Internal node analysis

Verify one of conditions $f_v^{x_i=1}(\gamma)=1$, $f_v^{x_i=0}(\gamma)=0$ and $f_v^{x_i=1}(\delta)=0$, $f_v^{x_i=0}(\delta)=1$. Let us verify the first condition. For that we execute the following steps.

Algorithm

1. Form the ROBDD implementing the function $f_v^{x_i=1}$ for the given internal node *v*. Call it as a ROBDD $(f_v^{x_i=1})$. Its root is the internal node in which the bold edge from *v* runs. The terminal nodes of the ROBDD $(f_v^{x_i=1})$ coincide with the terminal nodes of the ROBDD of *f*. When forming the ROBDD $(f_v^{x_i=0})$, we do the same. Its root is the internal node in which the dashed edge from *v* runs.

2. To get the ROBDD $(\overline{f}_v^{x_i=0})$ we rearrange the terminal nodes of the ROBDD $(f_v^{x_i=0})$.

3. Multiply the ROBDD $(f_v^{x_i=1})$ and the ROBDD $(\overline{f}_v^{x_i=0})$. Denote the result as a ROBDD R^*.

4. If the ROBDD R^* is not empty, we consider any path from the R^* root till its 1 terminal node. Note the corresponding product as *k*. The Boolean vector that turns *k* into 1 call the vector γ.

Note that the vector δ may be found in the similar way, that is, by multiplication of the ROBDD $(f_v^{x_i=0})$ and the ROBDD $(\overline{f}_v^{x_i=1})$. If the results of both multiplications are not empty we cover the internal node *v* with Invert-AND-OR sub-circuit.

Otherwise, one of the functions $f_v^{x_i=1}$, $f_v^{x_i=0}$ is implicant of another one. In that case, we cover the internal node *v* with Invert-AND-XOR sub-circuit.

Apply the suggested procedure for the ROBDD in Fig. 1. For right internal node marked x_2 we have:

$$f_v^{x_2=0} = x_3 x_4 \overline{x}_5 \vee x_3 \overline{x}_4 x_5 \vee \overline{x}_3, \qquad f_v^{x_2=1} x_4 \overline{x}_5 \vee \overline{x}_4 x_5 .$$

This means that $f_v^{x_2=1}$ is implicant of $f_v^{x_2=0}$. Consequently, we must cover the corresponding node with Invert-AND-XOR sub-circuit. For right internal node marked x_3 we have:

$$f_v^{x_3=0}=1, \qquad f_v^{x_3=1}=x_4\bar{x}_5\vee\bar{x}_4x_5 .$$

This means that $f_v^{x_3=1}$ is implicant of $f_v^{x_3=0}$. Consequently, we must cover the corresponding node with Invert-AND-XOR sub-circuit.

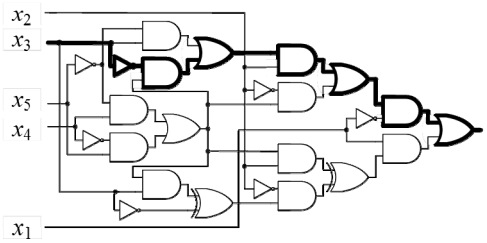

Fig. 4. Circuit C

For the rest internal nodes the results of both above mentioned multiplications are not empty. It means that the corresponding nodes may be covered with Invert-AND-OR sub-circuit. As a result of covering the circuit ROBDD we have combinational circuit *C* (Fig. 4).

Derive the formula from circuit *C* substituting the proper gate functions for circuit internal variables and eliminating brackets. Literal permutations and any simplifications are forbidden. In obtained formula some products are connected each other with operation OR ($\vee$) others – with operation XOR ($\oplus$). (As the products obtained in the course of substitutions are pairwise orthogonal, then the expressions like $(a\oplus b)\vee c$ is similar to the expression $a\oplus(b\vee c)$ and, consequently, brackets in both of them may be deleted). Call this formula as mixed SoP (MSoP). The formula is as follows:

$$MSoP(f)=x_1\bar{x}_2\bar{x}_3\oplus x_1\bar{x}_2x_3\bar{x}_4x_5\vee x_1\bar{x}_2x_3x_4\bar{x}_5\oplus x_1x_2\bar{x}_4x_5\vee$$
$$\vee x_1x_2x_4\bar{x}_5\vee\bar{x}_1\bar{x}_2\bar{x}_4x_5\vee\bar{x}_1\bar{x}_2x_4\bar{x}_5\vee\bar{x}_1x_2\bar{x}_3\bar{x}_4x_5\vee$$
$$\vee\bar{x}_1x_2\bar{x}_3x_4\bar{x}_5\vee\bar{x}_1x_2x_3\bar{x}_5.$$

3. MSoP properties

For any circuit obtained by the method discussed in Sections II, III we derive the system of MSoPs. Each literal of MSoP is connected with the path that begins at the circuit input marked by the input variable and terminates at the circuit output.

Consider the sub-circuit of Fig. 2 in detail. Call the sequence of gates with numbers 1, 2 and 4 as an inverse branch of the sub-circuit and the sequence of gates with numbers 3 and 4 as a direct branch of the sub-circuit. Each branch represents two paths of the sub-circuit. Examine two paths that begin at the input marked with the variable x_i and belong to the different branches. Call these paths as opposite ones. The path corresponding to the inverse branch is represented as literal $\bar{x}_i$ in the proper product of the MSoP and the path corresponding to the direct branch – as literal x_i. At the same time these paths correspond to dashed and bold edges of the ROBDD running from the node x_i that is covered by the sub-circuit when deriving circuit *C*. If the output of the sub-circuit is not output of circuit *C,* then the path of circuit *C* beginning at the input marked with the literal x_i ($\bar{x}_i$) is represented by the sequence of literals corresponding to the sub-circuits that are traversed by this path. This sequence is contained at least in one of the products of the MSoP. Moreover, this sequence is represented by the ROBDD path connecting the ROBDD root and the node x_i. The ROBDD path and the corresponding circuit *C* path are shown in Fig. 1 and Fig. 4.

For example, the path of circuit *C* marked with the bold line (Fig. 4) is represented as $\bar{x}_3x_2\bar{x}_1$. This sequence of literals is contained in the 8-th and 9-th products of the MSoP. Thus the literal position explicitly

determines the corresponding path in circuit *C*. Moreover, the sequence of literals is also represented by the ROBDD path (Fig. 1).

In [9] path delay faults are considered as temporary stuck-at faults of the corresponding ENF literals. This means that all literals satisfying to the same path are changed for the same constant. Instead of ENF of circuit *C* we may consider the MSoP created by circuit *C*. Note that for each literal of a MSoP product the brief information about the sequence is represented by the literals of this product forgoing to the considered literal.

In order to find all MSoP products that contain the literal x_i corresponding to the certain path α we need the following.

Find ROBDD path ε running from ROBDD root to node *v* marked with the literal x_i. The bold edge corresponding to the beginning of the path α runs from the node *v*. Call the corresponding to the path ε product as k_ε. Continue the ROBDD path ε through the bold edge to the 1 terminal node of the ROBDD. Find all prolongations and consequently all products containing the literal x_i corresponding to the beginning of the path α. These products comprise a set K_α [9]. Note that the products of the MSoP that contain the literal $\overline{x_i}$ corresponding to the path opposite to α have the same sub-product k_ε.

Let the literal x_i be equal to the constant 0 in each product of K_α.

Theorem 1. The Boolean vector β is a test pattern for the fault $x_i = 0$ in each product of K_α if it turns the certain product *K* from K_α into 1.

Proof. As all products of the MSoP are pairwise orthogonal then *K* is the only product of the fault-free MSoP that is turned into1on the vector β. When x_i is equal to 0 the fault MSoP is turned into 0 on the vector β. The theorem is proved.

Let the literal x_i be equal to the constant 1in each product of K_α.

Theorem 2. For the fault $x_i = 1$ in each product of K_α there exists the test pattern β.

Proof. Consider two cases.

The first case. The literal x_i corresponds to the input of Invert-AND-OR sub-circuit covering the internal node *v* marked with the variable x_i. In that case, there exists the test pattern γ that turns $f_v^{x_i=1}$ into 1 and $f_v^{x_i=0}$ into 0. Represent this test pattern with the product k_γ. Form the product $k_\varepsilon k_\gamma$. Let the product obtained from $k_\varepsilon k_\gamma$ by adding arbitrary (n–1) literals except x_i, $\overline{x_i}$ be k^*(γ originates k^*). Here *n* is a number of input variables of circuit *C*.

Let vector β be represented by the product $k^*\overline{x_i}$. This vector turns all products of K_α into 0. The rest of the products of fault-free MSoP are also turned into 0. When the MSoP is fault, that is, from each product of K_α the literal x_i is excluded, one of such product is turned into1on β. This is the product *K* of the MSoP corresponding to the function $f_v^{x_i=1}$ (the product *K* is originated by the product *k* of $f_v^{x_i=1}$ MSoP, which is turned into 1 on γ). $f_v^{x_i=1}$ MSoP is formed by the ROBDD whose root is the node in which a bold edge runs from *v*. The product *K* is the only one product as all products of the MSoP are pairwise orthogonal.

The second case. The literal x_i corresponds the input of Invert-AND-XOR sub-circuit covering the internal node *v* marked with the variable x_i. Consider the situation when there is no the test pattern γ that turns $f_v^{x_i=1}$ into 1 and $f_v^{x_i=0}$ into 0. Then there exists the test pattern *γ* that turns $f_v^{x_i=1}$ into 1 and $f_v^{x_i=0}$ into 1.

Form the vector β in the above mentioned way. It turns into 1 the certain product from the MSoP namely the product *K* originated by the $f_v^{x_i=0}$ MSoP (we take in mind the product *k* from the $f_v^{x_i=0}$ MSoP which is turned into 1 on γ). The vector β turns into 0 the rest products of the fault-free MSoP. Thus the vector β turns into 1 the fault-free MSoP.

When the MSoP is fault, that is, from each product of K_α the literal x_i is excluded, one of K_α products is turned into 1 on β (we take in mind the product *K* originated by the *k* from $f_v^{x_i=1}$ MSoP which is turned into 1 on γ). At the same time the vector β turns into 1 the product originated by $f_v^{x_i=0}$ MSoP (we refer to the product from the $f_v^{x_i=0}$ MSoP which is turned into 1 on γ). The vector γ turns into 1 only one product from $f_v^{x_i=1}$ MSoP and only one product from $f_v^{x_i=0}$ MSoP as all products of $f_v^{x_i=0}$ MSoP and $f_v^{x_i=1}$ MSoP are pairwise orthogonal.

Thus, two products are turned into 1 in the fault MSoP. The rest of the products are turned into 0 on the vector β. As we use Invert-AND-XOR sub-circuit for covering the internal node v marked with x_i, then for the node v and the fault considered we have the expression: k_ε ($f_v^{x_i=1} \oplus \overline{x}_i f_v^{x_i=0}$). After substitution of the vector β we get $1 \oplus 1$. This means that the vector β turns the fault MSoP into 0. Consequently, the vector β is a test pattern. The theorem is proved.

Thus, both faults of the *MSoP:* $x_i = 0$ in all products of K_α and $x_i = 1$ in all products of K_α are detectable.

4. Finding test pairs for robust testable and validatable non robust testable faults

Finding test pairs for robust testable and validatable non robust testable faults is based on the approach suggested in [7]. It is necessary to justify its application for the circuits that originate MSoPs instead of Reed-Muller expressions in [7]. In [9] the conditions of robust path delay fault manifestation for test pairs extracted from an ENF are formulated. The first condition is existence of a test pattern for the corresponding constant fault of the ENF literal. This test pattern is a vector v_2 of a test pair. The next condition is: the variable x_i that marks the node v takes the opposite values for v_1, v_2 vectors. Evidently, (Section 3) both conditions are feasible for the considered faults of MSoP.

Let $k(u)$ be minimal cube covering vectors v_1, v_2 of a test pair. Note as $\underline{K}$ the product that differs from the product K only by inversion of the variable x_i.

Remind that as we consider ROBDD, then for each node v the condition $f_v^{x_i=0} \neq f_v^{x_i=1}$ is executed. Let γ be a test pattern on which these functions are different and $f_v^{x_i=1}(\gamma) = 1$, $f_v^{x_i=0}(\gamma) = 0$. Represent this test pattern as we have done before with the product k_γ. Form the product $k_\varepsilon k_\gamma$ and k^* in the above mentioned way.

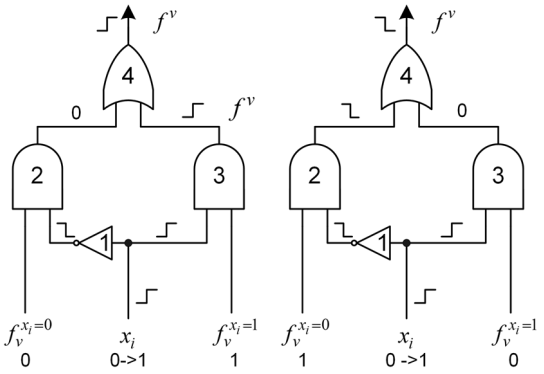

Fig. 5. Illustration of the corollary

Theorem 3. The product k^* represents the test pair of robust testable fault for the path α and its rising and falling transitions if the beginning of the path is marked with the literal x_i and the condition $f_v^{x_i=1}(\gamma) = 1$, $f_v^{x_i=0}(\gamma) = 0$ is executed.

Proof. The product $x_i k^*$ represents the Boolean vector v_2 that turns the product K from K_α into1. This means v_2 is a test pattern for the fault $x_i = 0$ in all products of K_α. Actually this vector turns $f_v^{x_i=0}$ MSoP into 0 and turns $f_v^{x_i=1}$ MSoP into 1. Moreover, this vector turns into 1 the only product K from K_α originated by $f_v^{x_i=1}$ MSoP. Consequently, this vector turns the fault–free MSoP into 1on the same product K and only on this product. As well this vector turns the fault MSoP into 0.

The product $x_i k^*$ represents the vector v_1 of the proper test pair. This vector is a test pattern for the fault $x_i = 1$ in all products of K_α. Actually the vector v_1 turns $f_v^{x_i=0}$ MSoP into 0 and turns $f_v^{x_i=1}$ MSoP into 0. Consequently, the vector turns the fault-free MSoP into 0. At the same time this vector turs into 1the fault MSoP because it turns into 1 the product $\underline{K}$ and the product K^* obtaind from K (K from K_α) when the considered fault occurs. The fault MSoP turns into 1 on the only product K^*. Thus the vector represented by product $x_i k^*$ is the

vector v_2 for rising transition of the path α and the vector represented by product $\bar{x}_i k^*$ is the vector v_2 for the falling transition of the path α. Note that $k(u)$ is orthogonal to all products of the MSoP except products of K_α. Actually $k(u)$ is orthogonal to the products of the MSoP that does not contain sub-product k_ε. The product $k(u)$ is also orthogonal to the products of the MsoP containing sub-product $k_\varepsilon \bar{x}_i$ as k_γ is orthogonal to $f_v^{x_i=0}$. Moreover, any product of the fault free MSoP does not contain repeated literals. Thus all conditions [9] of robust testable manifestation for rising and falling transition of the path α are fulfilled. The theorem is proved.

Corollary. If there exist a vector γ for which $f_v^{x_i=1}(\gamma)=1$, $f_v^{x_i=0}(\gamma)=0$ and a vector δ for which $f_v^{x_i=1}(\delta)=0$, $f_v^{x_i=0}(\delta)=1$, then for both paths that begin at the same input and marked with the literals x_i and $\bar{x}_i$ PDFs manifest themselves as robust testable for rising and falling transitions.

Fig. 5 is an illustration of the above mentioned corollary.

Note that sometimes only one condition pointed in the corollary is executed. Then the corresponding path has a test pair on which PDF manifests itself as robust testable one in both directions. This path must be tested first and then we proceed to test opposite path. Let opposite path be α. Let γ be a Boolean vector on which the condition $f_v^{x_i=1}(\gamma)=1$, $f_v^{x_i=0}(\gamma)=1$ is executed. The product k^* is formed in the above mentioned way.

Theorem 4. The product k^* represents the test pair of non robust testable PDF for the path α and its rising and falling transitions if the beginning of the path is marked with the literal x_i and the condition $f_v^{x_i=1}(\gamma)=1$, $f_v^{x_i=0}(\gamma)=1$ is executed.

Proof. The product $x_i k^*$ represents the Boolean vector v_2 that turns the product K from K_α into 1. This means v_2 is a test pattern for constant fault $x_i = 0$ in all products of K_α. The product $\bar{x}_i k^*$ represents vector v_1 of this test pair. The test pair detects rising transition of the path α. Theorem 2 should be taken to conclude that v_1 is a test pattern for the fault $x_i = 1$ in all products of K_α. The latter means that v_1 is as well as v_2 for the fault $x_i = 1$ in all products of K_α, that is, $\bar{x}_i k^*$ is a test pattern for falling transition. Note that $k(u)$ is orthogonal to all products of MSoP that does not contain sub-product k_ε. The product $k(u)$ is not orthogonal to some product K' containing k_ε and $\bar{x}_i$ as k_γ is not orthogonal to $f_v^{x_i=0}$. The theorem is proved.

Thus when the conditions of the above mentioned corollary is not executed we have robust testable PDF for the path that begins at the input marked with the variable x_i and validatable non robust testable PDF for the opposite path. Fig. 6 is illustration of robust testable PDF and validatable non robust testable PDF for paths that begin at the input marked with the variable x_i.

Remind that the product obtained from $k_\varepsilon k_\gamma$ by adding arbitrary (n–1) literals except x_i, $\bar{x}_i$, represents the test pair either for robust testable PDF or for validatable non robust testable PDF. In the first case, the vector γ, for which $f_v^{x_i=1}(\gamma)=1$, $f_v^{x_i=0}(\gamma)=0$, is represented by any path from ROBDD R* obtained with the algorithm of Section III. In the second case, the vector γ, for which $f_v^{x_i=1}(\gamma)=1$, $f_v^{x_i=0}(\gamma)=1$, is represented by any path of the ROBDD ($f_v^{x_i=1}$) (ROBDD ($f_v^{x_i=0}$)) that is an implicant of the ROBDD ($f_v^{x_i=0}$)(ROBDD ($f_v^{x_i=1}$)). The relation of implication is determined by the same algorithm. Note that the algorithm implementation has a polynomial complexity as the algorithm is based on multiplication of two ROBDDs extracted from the circuit ROBDD. Finding the vector γ was out of consideration in the paper [7].

As for k_ε it is originated by the path α connecting the circuit ROBDD root corresponding to the path output and the internal node v corresponding to the path input. It means that algorithm of finding test pairs has a polynomial complexity.

In order to detect validatable non robust PDF of the path and its rising and falling transitions, we must first deliver the test pair to detect robust testable PDF for the opposite path. If the opposite path is fault-free, we may detect validatable non robust PDF of rising and falling transitions of the considered path.

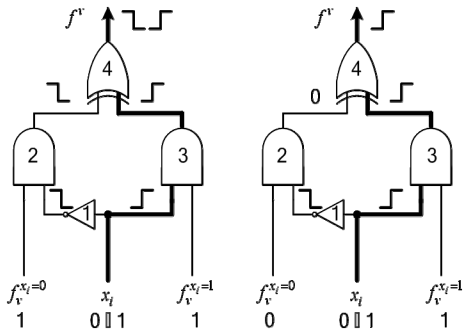

Fig. 6. An illustration of robust testable PDF (left)
and validatable non robust testable PDF (right)

5. Experimental results

In Table 1 the information about combinational circuits investigated (MCNC) is given. We have got representation of each combinational circuit by the Shared ROBDD using BuDDy system. For the Shared ROBDD the number of internal nodes are counted (second column of Table 2).

Table 1

Benchmark description

benchmark	inputs	outputs
C3540	49	22
C1908	33	25
C1355	41	32
C880	62	26
pair	173	137
frg2	143	139
k2	45	45
x3	135	99

Table 2

Benchmark results

benchmark	node count	or nodes	percentage	old area	new area	area reduction
C3540	672 435	513528	76,4%	5379480	3 325 368	39,2%
C1908	49 323	39238	79,6%	394584	237 632	39,8%
C1355	50 682	45393	89,6%	405456	223 884	44,8%
C880	346 688	225196	67%	2773504	1 872 720	32,5%
pair	51 414	790	1,5%	411312	408 152	0,8%
frg2	1542	79	5,1%	12336	12 020	2,6%
k2	698	341	48,9%	5584	4 220	24,4%
x3	543	38	7%	4344	4 192	3,5%

Then we found the nodes that may be covered by sub-circuit of Fig. 3 (third column of Table 2). The percentage of such sub-circuits among all ones was calculated (forth column of Table 2). We appreciated the complexity of the circuits obtained from the Shared ROBDD [9] by covering internal nodes with sub-circuit of Fig. 3 and the circuits obtained by the method suggested here. For that the numbers of two inputs and one input gates of the circuits are computed. The results are given in the sixth and seventh columns, correspondingly.

The last column of Table 2 describes the simplification of circuits on account of using the suggested method (in percentage terms). We see that always more simple circuits are derived and often essential simplification is possible.

Conclusion

New synthesis method of fully delay testable circuits is suggested that as a rule derives more simple circuits of this kind than in [9]. The simplification is based on multiplications of sub-circuit ROBDDs (these operations have the polynomial complexity). The circuits obtained originate new type of formulae called MSoPs in which products are separated either by symbol"⊕" or "∨". The investigation of the formulae properties insured formulating the algorithms of finding PDF test pairs that also have a polynomial complexity.

REFERENCES

1. Ashar, P., Devadas, S. & Keutzer, K. (1991) Gate-delay-fault testability properties of multiplexor-based networks. *Proc. Int. Test Conf.* pp. 887-896. DOI: 10.1007/BF01383945
2. Ashar, P., Devadas, S. & Keutzer, K. (1991) Testability properties of multilevel logicnetworks derived from binary decision diagrams. *Proc. Adv. Res. VLSI.* Univ. California, Santa Cruz. pp. 33–54.
3. Ashar, P., Devadas, S. & Keutzer, K. (1993) Path-delay-fault testability properties of multiplexor-based networks. *Integration, VLSI J.* 15(1). pp. 1-23. DOI: 10.1016/0167-9260(93)90002-T
4. Becker, B. (1998) Testing with decision diagrams. *Integration, VLSI J.* 26. pp. 5-20.
5. Drechsler, R., Shi, J. & Fey, G. (2004) Synthesis of fully testable circuits from BDDs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. 23(3). pp. 1-4. DOI: 10.1109/TCAD.2004.823342
6. Matrosova, A. & Nikolaeva, E. (2010) PDFs testing of combinational circuits based on covery ROBDDs. *Proceedings of EWDT Symposium.* pp. 160-163. DOI: 10.1109/EWDTS.2010.5742045
7. Matrosova, A., Nikolaeva, E., Kudin, D. & Singh, F. (2012) PDF testability of the circuits derived by special covering ROBDDs with gates. *Proceedings of EWDT Symposium.* Kharkov: IEEE.
8. Bryant, R.E. (1986) Graph-based algorithms for Boolean function manipulation. *IEEE Trans. on Computers.* pp. 677–691. DOI: 10.1109/TC.1986.1676819
9. Matrosova, A., Lipsky, V., Melnikov, A. & Singh, V. (2010) Path delay faults and ENF. *Proceedings of EWDT Symposium.* pp. 164-167.

Matrosova Anjela Yurievna, Dr. Science, prof. E-mail: mau11@yandex.ru
Mitrofanov Evgenii Vladimirovich. E-mail: qvaz@yandex.ru
Национальный исследовательский Томский государственный университет
Toral Shah. Shah@yandex.ru
Indian Institute of Technology (Bombay)

Поступила в редакцию 3 сентября 2016 г.

Матросова А.Ю., Митрофанов Е.В. (Томский государственный университет), *Шах Т.* (Индийский институт технологий Бомбей).

Упрощение контролепригодных комбинационных схем и поиск тестовых пар для неисправностей задержек путей.

Ключевые слова: робастно и не робастно тестируемые неисправности задержек путей; ROBDD-графы; контролепригодный синтез.

DOI: 10.17223/19988605/39/11

Разработан новый подход к синтезу тестопригодных комбинационных схем, в которых задержка каждого пути обнаружима, позволяющий по сравнению с известными ранее методами упростить структуры синтезируемых схем и сократить длины путей в них. Комбинационные схемы конструируются покрытием некоторого определенного подмножества вершин ROBDD графа Invert-AND-OR (НЕ, И, ИЛИ) подсхемами, реализующими формулу $f_v = \overline{x_i} f_v^{x_i=0} \vee x_i f_v^{x_i=1}$, и покрытием остальных вершин графа подсхемами Invert-AND-XOR (НЕ, И, И-ИЛИ), реализующими формулу $f_v = \overline{x_i} f_v^{x_i=0} \oplus x_i f_v^{x_i=1}$. Использование данного подхода позволяет сократить длины путей комбинационной схемы и уменьшить число OR, AND, NOT вентилей при условии реализации элемента XOR в базисе этих вентилей. Показано, что неисправности задержек путей в получаемых схемах проявляют себя либо как робастно тестируемые, либо как обнаружимые не робастно тестируемые. Доставляя тестовые наборы в определенном порядке, возможно обнаружить все неисправности задержек путей в схеме. Таким образом, предлагаемые в работе схемы являются стопроцентно тестируемыми относительно неисправностей задержек путей и не требуют введения дополнительного входа в схему. Экспериментальные данные показали, что данный метод позволяет существенно сократить сложность синтезируемых схем.