МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

ВЕСТНИК
ТОМСКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА

УПРАВЛЕНИЕ, ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И ИНФОРМАТИКА

TOMSK STATE UNIVERSITY
JOURNAL OF CONTROL AND COMPUTER SCIENCE

Научный журнал

2012 **№ 3(20)**

Свидетельство о регистрации: ПИ № ФС 77-29497
от 27 сентября 2007 г.

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Научный журнал «Вестник Томского государственного университета. Управление, вычислительная техника и информатика» был выделен в самостоятельное периодическое издание из общенаучного журнала «Вестник Томского государственного университета» в 2007 г. Журнал зарегистрирован в Федеральной службе по надзору в сфере массовых коммуникаций, связи и охраны культурного наследия (свидетельство о регистрации ПИ № ФС 77-29497 от 27 сентября 2007 г.), ему присвоен международный стандартный номер сериального издания (ISSN 1998-8605). С 2010 г. журнал входит в Перечень ВАК. Журнал выходит ежеквартально и распространяется по подписке, его подписной индекс 44031 в объединённом каталоге «Пресса России».

В журнале «Вестник ТГУ. УВТиИ» публикуются результаты теоретических и прикладных исследований вузов, научно-исследовательских, проектных и производственных организаций в области управления, вычислительной техники и информатики в технических, экономических и социальных системах.

Тематика публикаций журнала:

- УПРАВЛЕНИЕ ДИНАМИЧЕСКИМИ СИСТЕМАМИ
- МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ
- ОБРАБОТКА ИНФОРМАЦИИ
- ИНФОРМАТИКА И ПРОГРАММИРОВАНИЕ
- ДИСКРЕТНЫЕ ФУНКЦИИ И АВТОМАТЫ
- ПРОЕКТИРОВАНИЕ И ДИАГНОСТИКА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

Правила оформления статей приведены на сайте: **http://vestnik.tsu.ru/informatics/**

Адрес редакции:
634050, г. Томск, пр. Ленина, д.36, корп. 2, к. 201
Электронный адрес: http://vestnik.tsu.ru
Контактный тел./факс: (3822) 529-599
E-mail: vestnik_uvti@mail.tsu.ru

ООО «Издательство научно-технической литературы»
634050, Томск, пл. Новособорная, 1, тел. (3822) 533-335

Редактор *Т.С. Портнова*
Верстка *Д.В. Фортеса*

Изд. лиц. ИД № 04000 от 12.02.2001. Подписано к печати 10.09.2012.
Формат 70 × 100 $^{1}/_{16}$. Бумага офсетная. Печать офсетная. Гарнитура «Таймс».
Усл. п. л. 11,77. Уч.-изд. л. 13,18. Тираж 300 экз. Заказ № 43.

ПРОЕКТИРОВАНИЕ И ДИАГНОСТИКА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

УДК 004.312

A.Yu. Matrosova, E.A. Nikolaeva, S.A. Ostanin, V. Singh

ROBUST PDFS TESTING OF COMBINATIONAL CIRCUITS BASED ON COVERING BDDS[1]

A method of deriving test pair v_1, v_2 for robust path delay fault (PDF) of special combinational circuit is suggested. Circuit is obtained by covering binary decision diagram (BDD) with look up table (LUT) based configurable logic blocks (CLBs). It is found out that for each path of the circuit there exists a test pair v_1, v_2 on which delay fault manifests itself as robust. Triplets v_1, v_2, v_1 or v_2, v_1, v_2 detect delays of both rising and falling transitions of the same path. Integrating triplets of all paths we derive test T that detects any path delay fault of the circuit, single and multiple.

Keywords: *path delay fault* (*PDF*), *robust PDF*, *binary decision diagram* (*BDD*), *design for testability*, *FPGA.*

Delay testing has become very important problem with development of nanometer technologies. The objective of delay testing is to detect timing defects degrading the performance of a circuit. Path delay fault model is considered more preferable for detection of timing defects.

To observe delay defects, it is necessary to generate and propagate transitions in the circuit input. This requires application of a pair of vectors v_1, v_2. The first vector v_1 stabilizes all signals in the circuit. The second vector v_2 causes the desired transition on the input of a circuit. Take into account that delays of falling transition and rising transition along the same path from a primary input to a primary output in a circuit may be different. In the general case it is necessary a pair of vectors v_1, v_2 for each kind of transitions of a path. We will call a pair of vectors on which PDF manifests itself as PDF test pair (for the corresponding transition along the path). Single and multiple PDFs are distinguished.

In accordance with the conditions of fault manifestation single PDFs are divided into robust and non robust [1, 2]. PDF is robust if there is a test pair on which the fault manifestation does not depend on delays of other circuit paths.

PDF is non robust if a manifestation of the fault on a test pair is possible only when all other paths of a circuit are fault free.

It is very important to provide testability during circuit design. Circuits derived from BDDs as a rule are implemented with multiplexors. Their testability is investigated under different fault models [3–6] but the approaches suggested did not provide100% testability. In the paper [7] simple transformation of a circuit is suggested that guarantees

[1] This work was supported by grant «РФФИ № 11-08-92694-ИНД».

100% testability for both single stuck-at fault (SAF) and PDF models. The circuits are derived from BDDs with using multiplexors. The size of a circuit is directly proportional to the given BDD size. Optimization connected with variable ordering directly transfers to the circuit size. Disadvantage of this approach consists of using additional input.

In the paper [8] circuits are derived by covering BDDs with CLBs. In this paper it is revealed that each single stuck-at fault at the CLB pole is equivalent either single stuck-at fault of the proper internal node of BDD or 10 (01) faults of edges coming from the internal node. It is also determined that each single stuck-at fault at the CLB pole is detectable. It means that the circuit guarantees 100% testability under SAF. Moreover test for all multiple stuck-at faults may be derived directly from test for single stack-at faults. Test for multiple stuck-at faults is 2.5 times longer [8] than test for single stuck-at faults (at the average).

In this paper we show that circuit obtained from BDD by covering CLBs guarantees 100% testability for PDFs without an additional input. A size of the circuit is directly proportional to the given BDD size. Optimization connected with variable ordering directly transfers to the circuit size. Moreover the lengths of tests for all PDFs of circuits considered in this paper and the numbers of three input elements of these circuits as a rule less than ones in the circuits implemented with multiplexors [7]. In this paper in comparison with the paper [9] the experimental results are represented and proofs of the theorems are given.

In Section 2 a problem of deriving special combinational circuits is discussed. In Section 3 test pair is found on which PDF manifests itself as robust for rising and falling transitions. In Section 4 experimental results are given.

1. A combinational circuit design

It is well known that BDD is a directed acyclic graph based on using Shannon decomposition in each non terminal node ***v***:

$$f_v = \overline{x_i} f_v^{x_i=0} + x_i f_v^{x_i=1},$$

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

$$f_v^{x_i=1} = f_v(x_1,...,x_i = 1,...,x_n)).$$

Here f_v is the function corresponding to the node **v**, dashed edge points to $f_v^{x_i=0}$ and solid edge points to $f_v^{x_i=1}$. A BDD is called ordered if variables are encountered in the same order on all paths connecting the BDD root with a terminal node. A BDD is reduced if it does not contain either isomorphic subgraphs nor nodes so that $f_v^{x_i=0} = f_v^{x_i=1}$. Reduced and ordered BDD is a canonical representation of Boolean function for the chosen order of variables [10].

Any path that connects the BDD root with the 1 terminal node originates the product of the Disjoint Sum of Products (DSoP) of a function f that is represented with this BDD. DSoP is a sum of products in which any two product cubes don't intersect.

Let $F = \{f_1,...,f_m\}$, be the system of Boolean functions describing a combinational circuit behavior. Derive BDD using the same order of variables for each Boolean function from F. Join isomorphic subgraphs in the different BDDs. Combine BDDs 1 terminal nodes into one 1 terminal node and their 0 terminal nodes into one 0 terminal node.

Due to we obtain the graph with m roots and two terminal nodes. This graph jointly represents a system of m Boolean functions. It is called Shared BDD [10]. Without loss of generality we will consider further system with one function.

In Fig. 1 a BDD for one output Boolean function is shown. For each path connecting the BDD root with the 1 terminal node define the product of the DSoP. The DSoP of the function f is as follows.

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

Eliminate from BDD all edges connected with 0 terminal node and obtain the BDD representing a combinational circuit behavior. Call this BDD as Circuit BDD. Cover Circuit BDD with CLBs to get a combinational circuit executing that we use the following rules [8].

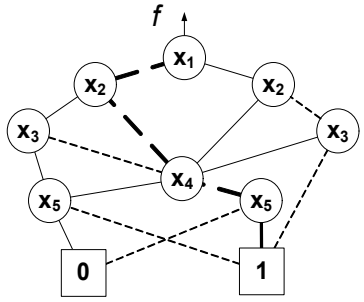

Fig. 1. BDD for f

1. CLB output corresponds to either non terminal node or the root of the Circuit BDD.

2. CLB input corresponds to either output of another CLB or variable of the Boolean function.

3. If two or more edges drop in a non terminal node of the Circuit BDD then this node may be split and covered with different CLBs.

4. The Boolean function implementing by a CLB is represented with the part (subgraph) of a Circuit BDD that is covered by the CLB. This function is derived from subgraph as DSoP depending on internal and input variables of the combinational circuit.

As a result we have got the combinational circuit C.

Covering Circuit BDD in accordance with rules 1–4 we have to provide coincidence of DSoPs system represented by the Circuit BDD with the DSoPs system derived from the combinational circuit C with substituting the proper DSoP of CLB instead of each internal variable of the combinational circuit C.

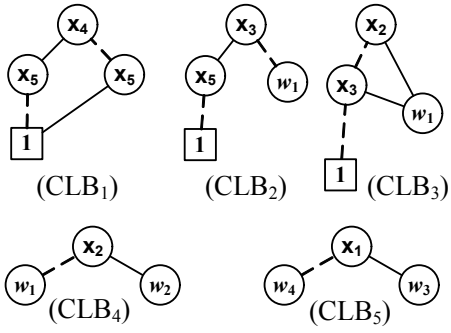

Fig. 2. Subgraphs of covering CLBs

Consider Circuit BDD obtained from the BDD in Fig. 1 with using 3 inputs CLBs. Subgraphs of covering CLBs are represented in Fig. 2. The circuit obtained is shown in Fig. 3.

Thus for each CLB we have subgraph of the Circuit BDD and the corresponding DSoP. CLB output may be either output of the circuit or internal node of the circuit that is input of another CLB.

For example DSoP $x_2w_1 \vee \overline{x}_2x_3w_1 \vee \overline{x}_2\overline{x}_3$ corresponds to CLB_3 derived from relevant subgraph of Fig. 2. Here x_2, x_3 – variables corresponding to the circuit inputs, and w_1 is internal circuit variable relating to the output of CLB_1.

Non terminal node of Circuit BDD corresponds to CLB output if the node is a root of the CLB subgraph.

2. Deriving test pair for PDF

2.1. Premises of test pair

We will examine one output circuit and corresponding BDD. Consider path α in the circuit of Fig. 3 (thick line). It begins from input variable x_4 and traverses the CLBs 1, 4, 5. CLB_5 output is output of the circuit.

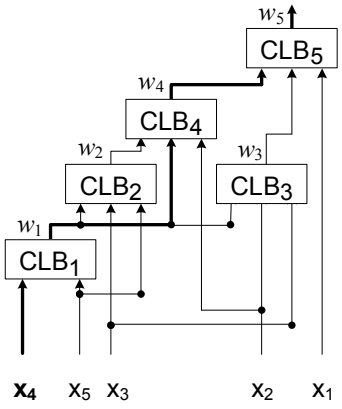

Fig. 3. Circuit *C*

We suppose that delays of the different paths from input to output of the same CLB are equal as CLB is LUT based. It means if a path α traverses certain CLB we may include any path of the CLB subgraph (connecting the subgraph root with its proper leaf) into the path α when investigating delay of the path α.

Derive reduced disjoint sum of products (reduced DSoP) for each CLB traversed by the path α. Reduced DSoP is formed from the CLB DSoP with including either products that contain the input variable corresponding to the beginning of the path α or products that contain the internal variable corresponding to the output of the previous CLB traversing with the path α.

For the chosen path α in Fig. 3 we have the following reduced and non reduced DSoPs (Fig. 2).

Reduced DSoP of CLB_1: $x_4\overline{x}_5 \vee \overline{x}_4 x_5$, non reduced DSoP of CLB_1 is the same.

Reduced DSoP of CLB_4: $\overline{x}_2 w_1$, non reduced DSoP of CLB_4: $x_2 w_2 \vee \overline{x}_2 w_1$.

Reduced DSoP of CLB_5: $\overline{x}_1 w_4$, non reduced DSoP of CLB_5: $x_1 w_3 \vee \overline{x}_1 w_4$.

Move along the path α from the output of the circuit to the beginning of the path α. Substitute reduced DSoPs of the corresponding CLBs instead of internal nodes traversed and remove brackets. If obtained sum of products has internal variables corresponding to the outputs of CLBs that are not traversed with the path α, then substitute instead of these variables the corresponding non reduced DSoPs and remove brackets till a set of internal variables becomes empty. Denote derived set of products as K_α.

Notice that we will consider any set of products here and further also as sum of these products (SoP).

For the path α in our example we have the result of the first substitution: $\overline{x}_1\overline{x}_2 w_1$. Then after the second substitution we have got K_α: $\overline{x}_1\overline{x}_2 x_4\overline{x}_5 \vee \overline{x}_1\overline{x}_2\overline{x}_4 x_5$. Take into consideration that if we would derive Equivalent Normal Form (ENF) from the circuit considered [12] then each literal of the sum of products would be supplied with sequence of numbers of elements corresponding to the proper circuit path.

Let products that contain literal ENF marked with the path α be called connected with the path α.

If we exclude in ENF sequences representing paths from all literals then K_α obtained above is a set of products connected with the path α.

Theorem 1. A set K_α contains all products connected with the path α. K_α is DSoP.

Proof. Each product from K_α contains either literal x_i or literal $\overline{x}_i$ that corresponds to the beginning of the path α and path α itself as each product is obtained with substitutions along the path α. By the construction K_α contains all products

corresponding to the path α. K_α is DSoP as changing in any DSoP some internal variable for corresponding DSoP originates also DSoP [13]. The theorem is proved.

Take into account that each BDD path connecting its root with 1 (0) terminal node originates the product. If the path traverses the internal node marked with *i* and solid (dashed) edge then the originated product contain x_i ($\overline{x}_i$).

Any path connecting two internal BDD nodes originates the product in the similar way.

For example the path of Fig. 1 (thick lines) originates product $\overline{x}_1\overline{x}_2\overline{x}_4x_5$ and part of this path connecting nodes 2 and 5 originates product $\overline{x}_2\overline{x}_4$.

Theorem 2. For each product from K_α there exists the path from the root of BDD to its 1 terminal node that originates this product.

Proof. From covering Circuit BDD with CLBs follows that we have got all products of DSoPs as a result of all substitutions of the proper CLB DSoPs instead of circuit internal variables. It means that we have got all products of DSoP for each function of the system. For the path α we execute part of substitutions, excluding internal variables to obtain all products connected with the path α. Consequently each product connected with the path α is among the BDD products corresponding to the one output circuit considered. The theorem is proved.

One test pattern from a test pair detecting robust PDF of both rising and falling transitions [14] has to turn into 1 product *K* from K_α possibly together with other products from K_α. In our case the circuit behavior is represented with DSoP. Consequently this test pattern turns into 1 only product *K*. This product contains either x_i, or $\overline{x}_i$.

Let $\underline{K}$ be obtained from *K* with changing literal $x_i(\overline{x}_i)$ for the inversion literal. Call $\underline{K}$ as an addition of *K*.

Another test pattern of a test pair has to turn into 1 $\underline{K}$ and into 0 DSoP derived from BDD [13] of the circuit considered. Denote this DSoP as D_d. Let *u* be minimal cube covering v_1 ,v_2 and k_u product representing *u*.

To detect robust PDF it is necessary to provide the condition: product k_u is orthogonal to products of D_d excluding product *K*.

Take into consideration that input variable x_i corresponding to the beginning of the path α is correlated (in general case) to several nodes of BDD covered by the CLB traversed with the path α. These nodes are marked with the same variable x_i. To find test pair we may choose any of them [8]. Denote the chosen node as ***v***.

For example if we would consider the path traversing the same CLBs that the path α, but beginning from the input variable x_5 we have got two nodes of BDD covered by CLB_1 and marked with the variable x_5.

Let ε be path from the BDD root into chosen node ***v***. It originates the product k_ε. Derive from K_α the products corresponding to the path ε that is products containing k_ε. Exclude from them variables of k_ε. Denote the result as K^*.

In the example considered a set of products corresponding to the path ε is as follows: $k_\varepsilon = \overline{x}_1\overline{x}_2$, $K^* = x_4\overline{x}_5 \vee \overline{x}_4x_5$.

Divide a set K^* into two subsets. Products of one subset have the literal x_i, and another – the literal $\overline{x}_i$. Exclude from them these literals and denote obtained subsets $K^*_{x_i}$, $K^*_{\overline{x}_i}$ correspondingly. These subsets represent functions implemented in nodes that

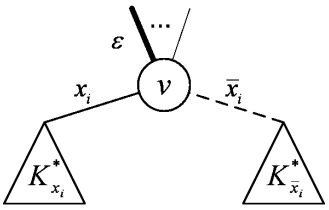

Fig. 4. Fragment of BDD

are incidental to right (solid) and left (dashed) edges of the node ***v*** (Fig. 4). These functions are different by the construction of BDD (in this paper we always means ROBDD)

In the example we have: $K^*_{x_i} = \bar{x}_5$, and $K^*_{\bar{x}_i} = x_5$.

Subsets $K^*_{x_i}$, $K^*_{\underline{x_i}}$ are determined with the pole **v** and do not depend on the chosen path ε (Fig. 4).

As the subsets represent different functions then there exists Boolean vector γ on which they take different values. Let γ turns into 1 $K^*_{x_i}$ and into 0 $K^*_{\underline{x_i}}$, k^* is a product representing γ.

In the example considered subsets $K^*_{x_i}$, $K^*_{\underline{x_i}}$ depend on the only variable x_5. Then we have: $\gamma = \overset{x_5}{0}$, $k^* = \bar{x}_5$.

First add literal x_i to k^* and then the product k_ε. Denote the result K', $K' = k^* x_i k_\varepsilon$ is absorbed by the only product K from K_α. It follows from the theorem 1 and the construction of Boolean vector γ. Let $\underline{K'}$ be addition of K' relative to x_i, then $\underline{K'}$ is absorbed by $\underline{K}$. It follows from the construction of Boolean vector γ and $\underline{K'}$. In the example: $K' = \bar{x}_1\bar{x}_2x_4\bar{x}_5$, $\underline{K'} = \bar{x}_1\bar{x}_2\bar{x}_4\bar{x}_5$. Here K, $\underline{K}$ coincide with K', $\underline{K'}$**.**

Two products are orthogonal if their cubes don't intersect.

A product k is orthogonal to the sum of products (SoP) if k is orthogonal to each product of the SoP.

Theorem 3. $\underline{K'}$ is orthogonal to D_d.

Proof. We have to show that $\underline{K'}$ is orthogonal to products of K^* that are obtained from K_α with using path ε in above mentioned way. Remind that $\underline{K'}$ contains subproduct k_ε. Product $\underline{K'}$ is orthogonal to set of products $K^*_{\underline{x_i}}$ under construction. Product $\underline{K'}$ is also orthogonal to set of products $x_i K^*_{x_i}$ hence $\underline{K'}$ is orthogonal to K^*. The theorem is proved.

Let Boolean vectors $\gamma^*, \underline{\gamma}^*$ turns into 1 products $K', \underline{K'}$ correspondingly and these vectors don't differ with variables that are absent in the products $K', \underline{K'}$. Let u be minimal cube covering these vectors and k_u – product representing this cube.

Theorem 4. Boolean vectors $\gamma^*, \underline{\gamma}^*$ comprise test pairs detecting robust PDF of the path α for both rising and falling transitions.

Proof. Vector γ^* turns into 1 product K (K from K_α) as K absorbs K' and vector $\underline{\gamma}^*$ turns into 1 product $\underline{K}$ as $\underline{K}$ absorbs $\underline{K'}$. Except $\underline{\gamma}^*$ turns into 0 D_d. Product k_u is orthogonal to products of D_d that does not contain k_ε as subproduct. Except (by the construction) k_u is orthogonal to a set of products $K^*_{\bar{x}_i}$ and a set of products $K^*_{x_i}$ without the product K. It means [13] that on vectors $\gamma^*, \underline{\gamma}^*$ PDF of α manifests itself as robust for both falling and rising transitions. The theorem is proved.

2.2. Algorithm of deriving test pair

Remind that each path connecting two nodes of BDD is related to the product originated by the path. Two paths are compatible if their originated products are not orthogonal.

The algorithm is partly (steps 1–5) based on results represented in [8] and connected with finding test pattern for single stuck-at fault of the circuit obtained by covering BDD with CLBs. We mean single stuck-at fault of the CLB pole directly connected with a circuit input.

1. Consider a path that begins from the node in which edge corresponding to x_i (solid edge) goes from the node **v** and ends into 1 terminal node of the BDD. Denote the path as η.

2. Look through paths that begin in the node in which edge corresponding to $\overline{x}_i$ (dashed edge) goes from the node **v** and ends into 0 terminal node of BDD in order to find path compatible with η. If we find such path then go to step 5. Otherwise return to step 1. If all paths η have looked through go to step 3.

3. Consider a path that begins from the node in which edge corresponding to $\overline{x}_i$ (dashed edge) goes from **v** and ends into 1 terminal node of BDD. Denote the path as η.

4. Look through paths that begin in the node in which edge corresponding to x_i (solid edge) goes from **v** and ends into 0 terminal node of BDD in order to find path compatible with η. If we find such path then go to step 5. Otherwise return to step 3.

5. Obtained path denote as ζ. Conjunction of products originated by the paths η, ζ, represents γ.

6. Derive Boolean vectors $\gamma^*, \underline{\gamma}^*$ in above mentioned way. These vectors have to turn into 1 the proper products $K', \underline{K'}$. Except, the Boolean vectors $\gamma^*, \underline{\gamma}^*$ do not differ with variables that are absent in products $K', \underline{K'}$.

In the example:

$$\gamma^* = \begin{matrix} x_1 x_2 x_3 x_4 x_5 \\ 0\ 0\ 0\ 1\ 0 \end{matrix},\ \underline{\gamma}^* = \begin{matrix} x_1 x_2 x_3 x_4 x_5 \\ 0\ 0\ 0\ 0\ 0 \end{matrix},\ K' = \overline{x}_1 \overline{x}_2 x_4 \overline{x}_5,\ \underline{K'} = \overline{x}_1 \overline{x}_2 \overline{x}_4 \overline{x}_5.$$

In the products $K', \underline{K'}$ variable x_3 is absent. In the Boolean vectors $\gamma^*, \underline{\gamma}^*$ this variable takes the 0 value.

In Fig. 5 thick lines represent paths corresponding to $\gamma^*, \underline{\gamma}^*$.

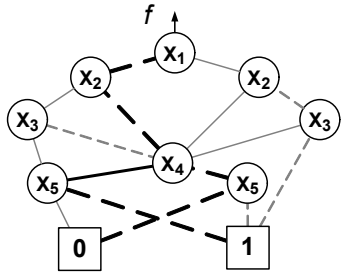

Fig. 5. Representing $\gamma^*, \underline{\gamma}^*$

To detect robust PDF of α for rising and falling transitions we need triplets v_1, v_2, v_1 or v_2, v_1, v_2. Integrating triplets of all circuit paths we derive test T for all path delay faults.

Theorem 5. Test T detects any PDF of a circuit (single and multiple).

Proof. As test T consists of pairs on which PDF manifests itself as robust and includes test pair for each path of a circuit then each single PDF is detectable with test T. As each single PDF manifests itself as robust on the proper pair of test T consequently any multiple PDF is detectable at the expense of single PDF comprising multiple fault. The theorem is proved.

Notice that all results derived for BDD easily may be spread to Shared BDD.

3. Experimental results

For the experiments we used the benchmarks LGSynth'91 [15].

In Table 1, the names of the benchmarks are given in the first column. The numbers of inputs and outputs are given in the second and the third columns, respectively.

In section MUX-map, the results are given for a direct mapping of BDDs by multiplexors as described in [6]. The number of nodes in BDD (NoN), the number of paths (NoP), and the PDF coverage (PDFC) are given in corresponding columns. This technique doesn't provide the 100% PDF covering. To provide 100% PDF covering for circuits in the frame of this technique it is necessary an additional input [7]. Using an additional input increases the number of nodes in BDD (and consequently the number of multiplexors) and the number of paths [7] for the same benchmark.

In section LUT-map, the results are given for the technique described above for 3 inputs CLBs. The number of CLBs in the circuit (NoC), the number of paths (NoP), and the PDF coverage (PDFC) are given in corresponding columns.

Experimental results

Name	in	out	MUX-map			LUT-map		
			NoN	*NoP*	*PDFC*	*NoC*	*NoP*	*PDFC*
5xpl	7	10	90	273	89.0	69	175	100.0
C17	5	2	12	22	68.1	6	12	100.0
alu2	10	6	259	873	86.9	246	929	100.0
b9	41	21	237	1773	64.6	141	380	100.0
clip	9	5	256	954	79.4	235	597	100.0
conl	7	2	20	47	74.4	9	18	100.0
count	35	16	251	2248	66.1	199	642	100.0
il	25	13	60	137	74.4	41	85	100.0
15	133	66	313	44198	61.3	169	941	100.0
t481	16	1	34	4518	86.1	26	1226	100.0
tcon	17	16	34	40	100.0	16	32	100.0
9sym	9	1	35	328	72.5	27	195	100.0
f51m	8	8	72	326	99.3	58	139	100.0
z4ml	7	4	66	175	77.1	50	118	100.0
x2	10	7	75	188	72.3	61	251	100.0

Experimental results showed that the number of CLBs and the number of paths are as a rule less then the number of multiplexers and the number of paths for the same benchmark.

Conclusion

Special combinational circuits are investigated. They are derived by covering BDDs with LUT based CLBs. It is found out that PDF of each path of such circuits manifests itself as robust. Except for delays of rising and falling transitions of the same path there exist triplets v_1, v_2, v_1 or v_2, v_1, v_2 detecting these delays. Triplets are found with BDD analyses based on finding test pattern for single stuck-at fault of CLB input directly connected with a circuit input. Experimental results showed that lengths of tests for PDFs of investigated circuits and the numbers of three input elements of these circuits

as a rule less than ones in circuits implemented with multiplexors. Investigated circuits do not demand additional input to provide 100% testability for robust PDFs.

REFERENCES

1. *Lin C.J.*, *Reddy S.M.* On Delay fault testing in logic circuits // IEEE Trans. on Computer-Aided Design. V. 6. No. 5. P. 694–701.
2. *Bushnell M. L.*, *Agrawal V. D.* Essentials of electronictesting for digital, memory and mixed-signal // VLSI Circuits. Hingham, MA, USA: Kluwer Academic Publishers, 2000. 432 p.
3. *Ashar P.*, *Devadas S.*, *Keutzer K.* Gate-delay-fault testability properties of multiplexor-based networks / P. Ashar, S. Devadas, K. Keutzer // Proc. Int. Test Conf. 1991. P. 887–896.
4. *Ashar P.*, *Devadas S.*, *Keutzer K.* Testability properties of multilevel logicnetworks derived from binary decision diagrams // Proc. Adv. Res. VLSI. Univ. California, Santa Cruz. 1991. P. 33–54.
5. *Ashar P.*, *Devadas S.*, *Keutzer K.* Path-delay-fault testability properties of multiplexor-based networks // Integration, VLSI J. 1993. V. 15. No. 1. P. 1–23.
6. *Becker B.* Testing with decision diagrams // Integration, VLSI J. 1998. V. 26. P. 5–20.
7. *Drechsler R.*, *Shi J.*, *Fey G.* Synthesis of fully testable circuits from BDDs // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 2004. V. 23. No. 3. P. 1–4.
8. *Matrosova A.*, *Lukovnikova E.*, *Ostanin S.*, *Zinchyk A.*, *Nikolaeva E.* Test generation for single and multiple stuck-at faults of a combinational circuit designed by covering shared ROBDD with CLBs // Proc. of the 22nd IEEE Intern. Symp. 2007. P. 206–214.
9. *Matrosova A.*, *Nikolaeva E.* PDFs testing of cmbinational circuits based on covery ROBDDs // Proc. of EW&DT Symposium. 2010. P. 160–163.
10. *Bryant R.E.* Graph-based algorithms for Boolean function manipulation // IEEE Trans. Comput. 1986. V. C-35. P. 677–691.
11. *Minato S.*, *Ishiura N.*, *Yajima S.* Shared binary decision diagram with attributed edges for efficient Boolean function manipulation // Proc. 27th IEEE/ACM DAC. 1990. P. 52–57.
12. *Armstrong D.B.* On finding a nearly minimal set on fault detection tests for combinational logic nets // IEEE Trans. Electronic Computers. 1966. EC-15. P. 66–73.
13. *Matrosova A.* Random simulation of logical circuits // Automation and Remote Control. 1995. No. 1. P. 156–164.
14. *Matrosova A.*, *Lipsky V.*, *Melnikov A.*, *Singh V.* Path delay faults and ENF // Proc. of EW&DT Symposium. 2010. P. 164–167.
15. *Yang S.* Logic synthesis and optimization benchmarks user guide // Tech. Rep., Microelectron. Center of North Carolina. 1991. 44 p.

Matrosova Anzhela Yu., *Nikolaeva Ekaterina A.*, *Ostanin Sergey A.*
Tomsk State University
Singh Virendra
Indian Institute of Technology Bombay
E-mail: mau11@yandex.ru; ostanin@mail.tsu.ru;
nikolaeve-ea@yandex.ru; virendra@computer.org

Поступила в редакцию 15 декабря 2011 г.

Матросова А.Ю., *Николаева Е.А.*, *Останин С.А.* (Томский государственный университет), *Сингх В.* (Индийский институт технологий, Бомбей). **Построение тестов для неисправностей задержек робастно тестируемых путей для комбинационных схем, построенных покрытием BDD-графов.**

Ключевые слова: неисправность задержки пути, робастно тестируемый путь, бинарные решающие диаграммы, контролепригодное проектирование, ПЛИС.

При тестировании неисправностей задержек путей особенно важно обнаружение робастно тестируемых путей. К сожалению, не все пути в произвольных схемах являются робастно тестируемыми. Установлено, что неисправность задержки каждого пути схемы, полу-

ченной покрытием системы ROBDD-графов программируемыми логическими блоками с сохранением системы ОДНФ (ортогональных дизъюнктивных нормальных форм), представляемой графами, проявляется как робастная. Предложен алгоритм построения пары тестовых наборов, обнаруживающей робастно тестируемую неисправность задержки пути. Найденная пара может быть использована для тестирования обоих перепадов значений сигналов пути при перестановке элементов пары. Тест, обнаруживающий робастно тестируемые неисправности задержек всех одиночных путей, обнаруживает все кратные неисправности задержек путей схемы и одиночные константные неисправности на полюсах логических элементов схемы.