Robert CZERWIŃSKI, **Tomasz RUDNICKI**
SILESIAN UNIVERSITY OF TECHNOLOGY, FACULTY OF AUTOMATICS, ELECTRONICS AND COMPUTER SCIENCE,
ul. Akademicka 16, 44-100 Gliwice

# An Algorithm of the Test Pairs Minimization by means of the Incompatibility Graph

**Dr inż. Robert CZERWIŃSKI**

He received M. Sc. degree and Ph. D. degree in Technical Sciences from the Faculty of Automatics, Electronics and Computer Science, Silesian University of Technology, Poland, in 2001 and 2006, respectively. He works in the Institute of Electronics, The Silesian University of Technology, Poland, as an Assistant Professor. His research interests include: programmable logic devices, logic synthesis and optimization.

*e-mail: robert.czerwinski@polsl.pl*

**Dr inż. Tomasz RUDNICKI**

He received M. Sc. degree and Ph. D. degree in Technical Sciences from the Faculty of Automatics, Electronics and Computer Science, Silesian University of Technology, Poland, in 2001 and 2006, respectively. He works in the Institute of Electronics, The Silesian University of Technology, Poland, as an Assistant Professor. His research interests include: diagnostic and design circuit for testing and high resolution ADC.

*e-mail: tomasz.rudnicki@polsl.pl*

**Abstract**

A heuristic method of the test pairs minimization in Two-Pattern testing is presented. The method is designed for test pattern generators including ROM and MISR, while the goal of the minimization is reduction of the ROM size. The method is based on the coloring the incompatibility graph. Authors present original application of the coloring the incompatibility graph. Introduced in the paper algorithm is very compact and can be implemented as a quick computer program. Primary experiments prove the high effectiveness of the method.

**Keywords**: two-pattern testing, delay faults, MISR.

## Algorytm minimalizacji par testowych z użyciem grafu niezgodności

**Streszczenie**

W niniejszej pracy przedstawiono heurystyczną metodę minimalizacji liczby par testowych potrzebnych do testowania uszkodzeń opóźnieniowych. Metoda ta polega na tworzeniu w kolejnych etapach minimalizacji grafu niezgodności par testowych. Minimalizacja opiera się na kolorowaniu takiego grafu. Ostateczna liczba par testowych jest równa liczbie chromatycznej grafu. Naturalnie, kolory przyporządkowane poszczególnym wierzchołkom grafu zawierają informację, które pary testowe mają być ze sobą sklejane. O końcowej liczbie par testowych po procesie sklejania decyduje liczba stanów nieokreślonych występujących w parach testowych przed procesem sklejania. Jeżeli liczba tych stanów jest duża, wówczas istnieje wiele możliwości sklejania par testowych. Jednak tylko kilka rozwiązań sklejania daje minimalną końcową liczbę par testowych po procesie sklejania. Metoda nie wymaga rozwiązania problemu pokrycia znanego z klasycznych metod minimalizacji. Kilka sklejonych par testowych (rys. 3) w jedną parę testową (rys. 4) oznacza, że w jednym takcie zegarowym zostanie przetestowanych kilka ścieżek układu ze względu na występowanie w nich uszkodzeń opóźnieniowych. Mniejsza liczba par testowych oznacza mniejszą liczbę słów programujących, a także mniejsze wykorzystanie pamięci ROM generatora par testowych z pamięcią ROM (rys. 1) przy jednoczesnym wysokim współczynniku pokrycia par testowych. Dodatkową zaletą mniejszej liczby par testowych jest mniejsza liczba potrzebnych taktów zegarowych do ich generacji. Poszczególne kroki metody minimalizacji liczby par testowych (rys. 6) zostały przedstawione na prostym przykładzie (rys. 5). Wstępne wyniki eksperymentów dają bardzo dobre wyniki.

**Słowa kluczowe**: testowanie, pary testowe, uszkodzenia opóźnieniowe, MISR.

## 1. Introduction

With a growing number of transistors on-chip, the problem of manufacturing defects becomes more serious. A testing phase in the production process is critical. In some cases test pattern generators are built based on the read only memory [3, 4, 5] (ROM). A ROM-based test pattern generator is presented in Fig. 1.

A counter is used for addressing the ROM, while control words saved in the ROM control the cycle of the multiple input signature register (MISR). Control words are given one by one to the inputs of the MISR in order to generate all possible test pairs to the circuit under test (CUT).

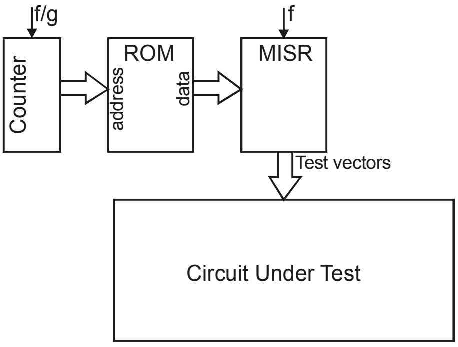

Fig. 1. A ROM-based test pattern generator
Rys. 1. Generator par testowych wykorzystujący pamięć ROM

The essence of presented testing method is appropriate selection of the test pairs to test all paths from primary inputs to primary outputs, and to determine the set of MISR’s control words. The advantage of presented structure is its scalability and independence from the CUT functions. Any change in the CUT function does not imply changes in the structure of the test pattern generator. The only change is that associated with the control words stored in the ROM.

To save the ROM utilization and to fasten the tests the procedure of control words merging is carried out [5]. To generate all possible test pairs, all control words must be used. Of course there is connection between the number of control words and the number of test pairs. The lower is the number of test pairs, the lower is the number of control words stored in the ROM and the quicker is testing process. Test pairs usually include many don’t cares, so in many cases the number of test pairs and indirectly control words may be reduced. Quick and easy to implement algorithm of the number of test pairs minimization, based on the coloring of the incompatibility graph, is addressed. The number of test pairs after minimization equals the chromatic number of the incompatibility graph.

## 2. Two-pattern testing

One of the basic test methods is Two-Pattern testing [1]. Delay faults and transistor stuck-open faults require the application of a two-vector sequence {U,W} in order to be detected.

The first input vector U initializes the circuit, while the second one W is required to launch the appropriate transition 0→1 or 1→0 and propagate the fault effect to the circuit under test output.

If every test pair {U,W} is implemented separately, two control words of MISR and 2·g clock cycles are required to generate these two pairs. Generally, test pairs include many don't-cares. In many cases a set of test pairs can be merged into one test pair. Let us consider the following example.

## Example

Schematic of an example combinatorial circuit is presented in Fig. 2.

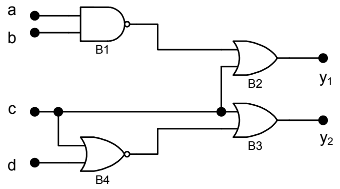

Fig. 2. Schematic of an example combinatorial circuit
Rys. 2. Schemat przykładowego układu kombinacyjnego

Two paths are to be tested paying special attention to delay faults in rising edge in outputs $y_1$ and $y_2$: path a-B1-B2-$y_1$ and path d-B4-B3-$y_2$. To test those paths two pairs of vectors must be calculated. First vector is used to adjust output in low level, while the second vector is used to generate rising edge on the output. One pair is for each path. Example of testing discussed two paths is presented in Fig. 3.

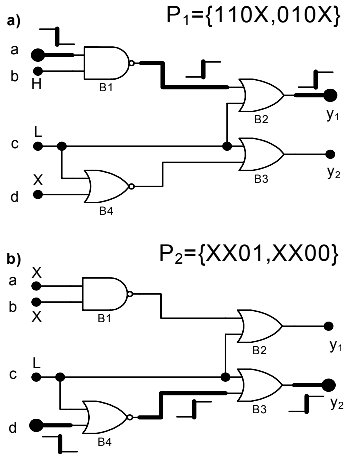

Fig. 3. Two pattern testing of an example combinatorial circuit
Rys. 3. Testowanie przykładowego układu kombinacyjnego z wykorzystaniem par testów

To test path a-B1-B2-$y_1$, the vector $U_1$, which enables to adjust the output $y_1$ to low logic level, is 110X. The vector which enables to generate rising edge on the output $y_1$ is 010X. So first test pair is $P_1$={110X,010X} (Fig. 3a).

Similarly, to adjust $y_2$ to low logic level in the path d-B4-B3-$y_2$, the input vector is XX01. The vector which enables to generate rising edge on the output $y_2$ is XX00, and the second test pair is $P_2$={XX01,XX00} (Fig. 3b).

It is possible to test both paths simultaneously, like in Fig. 4, because test pairs $P_1$ and $P_2$ can be merged into one test pair $P_{1,2}$={1101,0100}.

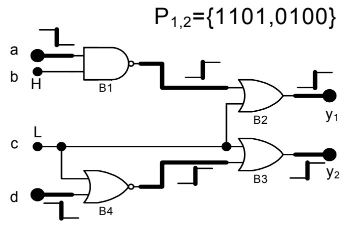

Fig. 4. Simultaneous testing of two paths in an example circuit
Rys. 4. Jednoczesne testowanie obydwu ścieżek przykładowego układu

To test discussed two paths independently, four clock cycles of the test pattern generator are required, while in simultaneous testing only two clock cycles are required.

## 3. An Algorithm

First of all, the incompatibility graph G(V,E) is creating, where: V is the vertex set of G denoted by V(G) and E is the edge set of G denoted by E(G). V(G) corresponds to a set of test pairs, where one vertex corresponds to one and only one test pair. E(G) corresponds to set of incompatible test pairs. Two test pairs (vertices) $P_1=\{u_1,w_1\}$ and $P_2=\{u_2,w_2\}$ are incompatible if, and only if in the ordered set of input vectors $\{u_1,u_2\}$ and $\{w_1,w_2\}$ exist at least one input for which exists pair (0,1) or (1,0). For example two test pairs $P_1$={**1**0X,100} and $P_2$={**0**0X,100} are incompatible because there is input (bolded) for which there exists pair (1,0).

The essence of the method of coloring the incompatibility graph G(V,E) is to assign colors to vertices in such a manner that every two adjacent vertices (sharing the same edge) are assigned to different colors. There are many different algorithms of a graph coloring. In a proposed approach authors based on quick and easy algorithms presented in [2, 6].

The proposed algorithm is as follows. Specific vertex $v_i$ is chosen in the $i^{th}$ step of the algorithm. The vertex is assigned with available color denoted by capital letter. Every adjacent vertex to the vertex $v_i$, is assigned with unavailable color. After coloring vertex $v_i$, the reduction of the incompatibility graph G(V,E) is carried out. This reduction consists in removing every edge from vertex $v_i$.

A selection of the vertex in the $i^{th}$ step of the algorithm is based on the principles:

- vertex with maximum number of unavailable colors is chosen,
- vertex with maximum number of edges is chosen.

In the next step of the algorithm, the vertex $v_{i+1}$ from reduced graph is chosen. The number of test pairs after minimization equals the chromatic number of the incompatibility graph. An algorithm of test pairs minimization is depicted in the example.

## Example

An example of test pairs and incompatibility graph is presented in Fig. 5.

P1={XX1XXX,X10XXX}
P2={X0XXX0,X10XX0}
P3={X0XXX0,X00XX1}
P4={X10XXX,X11XXX}
P5={X10XXX,X00XX0}
P6={XX0XX1,X00XX0}
P7={XXXX1X,X0X100}
P8={1XX0XX,10X100}

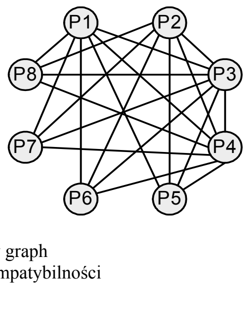

Fig. 5. Set of test pairs with incompatibility graph
Rys. 5. Zbiór par testowych z grafem niekompatybilności

In first step of the graph coloring a vertex with the greatest number of edges is chosen. Color 'A' is assigned to the vertex U3. To every vertex, which share edge with vertex U3, is assigned unavailable color labeled as 'a' and every edge connected to vertex U3 is canceled. The incompatibility graph after first step of coloring is presented in Fig. 6a.

As a next, a vertex U4 is colored. U4 is assigned a color 'B' because color 'A' is unavailable for vertex U4 (Fig. 6b). In the third step of the graph coloring, a vertex U1 is assigned a color 'C' (Fig. 6c). In the next stage, a vertex U5 is picked, according to rules: vertex with maximum number of unavailable colors is chosen and then vertex with maximum number of edges is chosen. A vertex U5 receives color 'D', because 'A', 'B' and 'C' are unavailable.

After that, a vertex U2 is colored because it has the same number of unavailable colors as vertices U6, U7, U8 but it has the greatest number of edges (Fig. 6d). Color 'C' is not unavailable for vertex U2, so 'C' is assigned to U2 (Fig. 6e). Vertices U6, U7 and U8 are assigned available color 'D' (Fig. 6f). Effects of the graph coloring with minimized test pairs are presented in Fig. 6f.

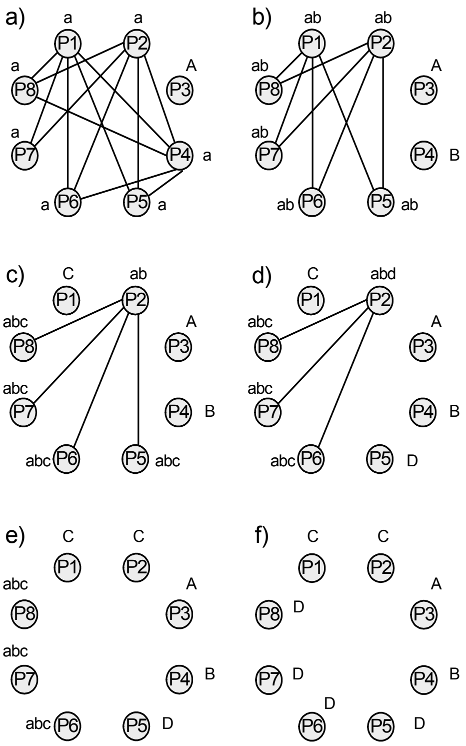

Fig. 6. Successive stages of the incompatibility graph coloring
Rys. 6. Kolejne iteracje kolorowania grafu niezgodności

Four colors are used to color the incompatibility graph (A, B, C, D), so eight test pairs presented in Fig. 5 are minimized into four pairs presented in Fig. 7.

$P_A$={X0XXX0,X00XX1}
$P_B$={X10XXX,X11XXX}
$P_C$={X01XX0,X10XX0}
$P_D$={110011,100100}

Fig. 7. Test pairs after the minimization
Rys. 7. Pary testowe po minimalizacji

## 4. Conclusions

This original heuristic method of minimizing the number of test pairs is used for the test pairs' reduction needed for delay faults. This method depends on creating in other stages incompatibility graph of the test pairs. After each stage the simplification of the incompatibility graph and coloring of the chosen vertex follows in the method. Using this method, we get the information of which test pairs can be merged together. About the final number of the test pairs after the merging process decides the number of don't cares presented in the test pairs before merging. If there are a lot of don't cares then there are a lot of possibilities of merging the test pairs, and in the same time there is a large number of the test pairs' reduction. However, only a couple of solutions of merging gives the minimal of the final number of the test pairs' merging. The method presented in this paper enables to get a minimize number of the merging test pairs. A number of the test pairs merged in one test pair means, that in one clock cycle are tested a number of circuit paths because of the presence of the delay faults in them. The lower number of test pairs means the lower number of the control words, and also the lower usage of the ROM generator for the test pairs. This gives a high coverage of the test pairs. Another advantage of the lower number of the test pairs is the lower number of clock cycles needed to their generation.

Presented in the paper algorithm is very compact and can be easily implemented as a quick computer program. Preliminary experiments prove high effectiveness of the algorithm. Introduced method of incompatibility graph coloring may be used as a minimization method of the control words in ROM-based test pattern generators for further reduction of the ROM size.

## 5. References

[1] Furuya K., McCluskey E.J.: Two-Pattern Test Capabilities of Autonomous TPG Circuits. Proc. IEEE Int. Test Conference, October 1991, pp. 704-711.

[2] Kania D.: Synteza logiczna przeznaczona dla matrycowych struktur programowalnych typu PAL. Zeszyty Naukowe Politechniki Śląskiej, Gliwice, 2004.

[3] Polian I., Becker B.: Configuring MISR-Based Two-Pattern BIST Using Boolean Satisfiability. Proc. Int. Workshop on Design and Diagnostics of Electronic Circuits and Systems, April 2003, pp. 73-80.

[4] Qiang X., Nicolici. N.: DFT infrastructure for broadside two-pattern test of core-based SOCs. IEEE Transactions on Computers, April 2006, Volume 55, Issue 4, pp. 470 – 485.

[5] Rudnicki T., Hławiczka A.: Test Pattern Generator for Delay Faults. Proc. of IEEE Int. Workshop on Design & Diagnostic of Electronics Circuits & Systems, DDECS'07, Kraków 2007, pp. 255-258.

[6] Wan W., Perkowski M.A.: A new Approach to the Decomposition of Incompletely Specified Multi-Output Functions Based on Graph Coloring and Local Transformations and Its Applications to FPGA Mapping. Proc. of the Conf. on European Design Automation, 1992, pp. 230-235.

*otrzymano / received: 04.01.2010*
*przyjęto do druku / accepted: 04.05.2010* *artykuł recenzowany*