

Tomasz RUDNICKI

POLITECHNIKA ŚLĄSKA, INSTYTUT ELEKTRONIKI

## Zmodyfikowany generator par testowych dla uszkodzeń opóźnieniowych

Dr inż. Tomasz RUDNICKI

W 2001 roku ukończył studia magisterskie na Wydziale Automatyki, Elektroniki i Informatyki Politechniki Śląskiej na kierunku Elektronika i Telekomunikacja. W 2006 roku obronił pracę doktorską. Od 2006 roku adiunkt w Instytucie Elektroniki Politechniki Śląskiej. Jego zainteresowania naukowe dotyczą testowania układów cyfrowych, projektowania przyrządów pomiarowych o dużej rozdzielczości oraz systemów mikroprocesorowych.



e-mail: tomasz.rudnicki@polsl.pl

### Streszczenie

W artykule przedstawiono metodę generacji par testowych pobudzających uszkodzenia opóźnieniowe. Źródłem par testowych jest zmodyfikowany rejestr MISR. Modyfikacja rejestru MISR polega na podwojeniu jego długości. Dzięki temu udało się ograniczyć do jednej liczby słów programujących, a tym samym zrealizować generator par testowych bez jakiegokolwiek pamięci. To spowodowało, że uzyskano podobne rezultaty jak dla generatora par testowych z pamięcią ROM, co jest główną zaletą przedstawionego generatora par testowych.

**Słowa kluczowe:** generator par testowych, uszkodzenia opóźnieniowe, rejestr MISR, pary testowe

### Modification of Test Pattern Generator for Delay Faults

#### Abstract

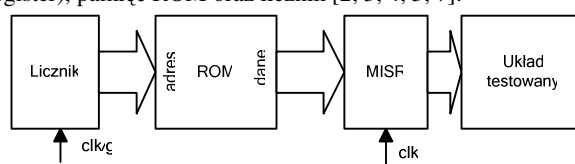
A method of generating test pairs for the delay faults is presented in this paper. The modification of the MISR register gives the source of test pairs. The modification of this register depends on doubling its length (fig. 3). Test pairs are only generating a half of chosen outputs of the MISR register. Doubling the MISR register makes possible to generate all possible test pairs, which was proved in the article [2, 3, 4]. The disadvantage of this result is too large number of clock cycles. Test pairs for the delay faults consists of a quite number of don't cares. This enables for a considerable reduction of test pairs. Minimizing the number of test pairs means smaller number of clock cycles with a very high coverage factor of test pairs. The process of merging test pairs is shown in the example. These modification result in only one number of programming words and in consequences enable to produce the generator of test pairs without ROM. Experimental results, in which the method of test pairs for benchmarks of the ISCAS'89, are presented in this paper. The choice of these benchmarks was limited by the benchmarks that had the number of inputs below 32. These results are similar to results of the generator of test pairs with ROM [1, 2, 4] (fig. 1). The coverage factor is situated somewhere between 65% and 95%, but with the sequence length ranging from 160, to 300k clock cycles. The main advantage of this solution is the lack of ROM.

**Keywords:** TPG, Delay Faults, MISR, Test Pairs

## 1. Wprowadzenie

Wraz ze stałym wzrostem liczby tranzystorów zawartych w jednej obudowie pojawiają się większe niedokładności technologii wytwarzania takich układów scalonych. Defekty takie jak: upływność pomiędzy wyprowadzeniami, zanieczyszczenia powierzchniowe, wilgoć mogą prowadzić do powstania uszkodzeń opóźnieniowych (ang. delay faults) [8, 9, 10]. Defekty tego typu mogą także powstawać podczas normalnej pracy układu na skutek jego starzenia się. Uszkodzenia opóźnieniowe do wykrycia wymagają podania pobudzeń testowych przy pełnej szybkości zegara. Obecnie jednak typowe testery zewnętrzne pracują z częstotliwościami wielokrotnie mniejszymi od tych stosowanych wewnątrz układów scalonych. W związku z tym konieczne jest wykorzystanie wbudowanego samotestowania BIST (ang. Built-In Self-Test) [2, 3].

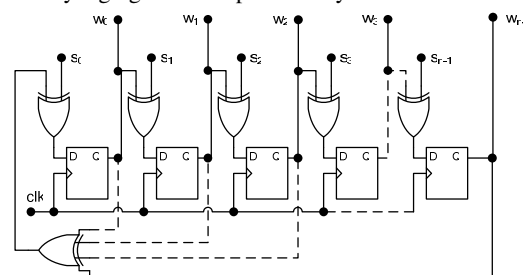
Do wykrywania uszkodzeń opóźnieniowych stosuje się wymuszenia testowe w postaci par złożonych z dwóch wektorów  $P=\{V_1, V_2\}$ . Wektor  $V_1$  to wektor inicjalizujący testowany układ cyfrowy CUT, natomiast wektor  $V_2$  to wektor generujący pożądany rodzaj zmiany stanu wybranego wyjścia układu testowanego (01 lub 10). Pożądany rodzaj zbrocza pobudza ewentualne uszkodzenia opóźnieniowe i tworzy warunki do propagacji tych pobudzonych uszkodzeń do wyjścia układu testowanego (ang. Circuit Under Test). Taka metoda testowania jest powszechnie znana jako testowanie dwu wektorowe (ang. Two-Pattern Testing - TPT) [1]. Na rysunku (rys. 1) przedstawiono generator par testowych zbudowany w oparciu o rejestr MISR (ang. Multi-Input Signature Register), pamięć ROM oraz licznik [2, 3, 4, 5, 7].



Rys. 1. Struktura generatora par testowych z pamięcią ROM

Fig. 1. Structure of Test Pattern Generator with ROM

Na rysunku (rys. 2.) przedstawiono strukturę rejestru MISR użytego do budowy tego generatora par testowych.

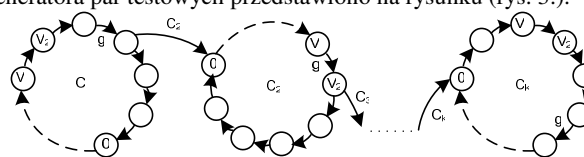


Rys. 2. Struktura rejestru MISR

Fig. 2. Structure of MISR register

Bezpośrednim źródłem par testowych dla układu testowanego CUT jest właśnie rejestr MISR. Do przechowywania  $L_s$ -bitowych słów programujących pracę rejestru MISR zastosowano pamięć ROM. Licznik służy do adresowania pamięci. Słowa programujące podawane są na wejścia  $SP=\{s_0, s_1, s_2, \dots, s_{n-1}\}$ . Jedynka logiczna występująca na pozycji bitu  $s_i$  odpowiada bramce negacji między wyjściem przerzutnika danej sekcji a wejściem przerzutnika kolejnej sekcji. Pary testowe generowane są na wyjściach rejestru MISR  $W=\{w_0, w_1, w_2, \dots, w_{n-1}\}$ .

Zaletą tej struktury jest skalowalność i niezależność od funkcji użytkowych układu CUT. W praktyce oznacza to, że wraz ze zmianą funkcji CUT struktura przedstawionego generatora par testowych nie zmienia się. Zmianie ulegają jedynie słowa programujące zapisane w pamięci ROM. Natomiast kolejność występowania słów w pamięci ROM nie ma znaczenia. Graf pracy takiego generatora par testowych przedstawiono na rysunku (rys. 3.).



Rys. 3. Graf pracy generatora par testowych z pamięcią ROM

Fig. 3. The operation diagram of the TPG

Możliwości stosowania tej struktury do generacji par testowych były badane w [2,3]. W pracach tych założono, że sprzężenie liniowe n-bitowego rejestru MISR zbudowane jest w oparciu o

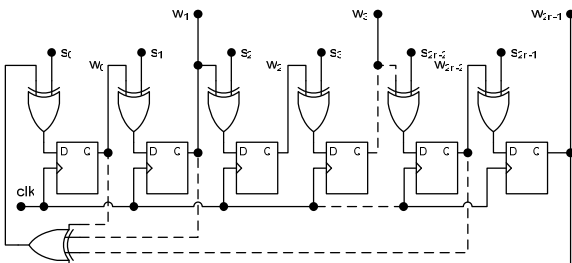
wielomian pierwotny gwarantujący  $2^{n-1}$  taktowy cykl pracy tego rejestru. Jednakże, dzięki licznikowi taktowanemu  $g$  razy wolniej niż rejestr MISR, ten ostatni pracuje w przypadku każdego słowa programującego  $C_i$  ze skróconym cyklem pracy, zawierającym tylko  $g < 2^{n-1}$  taktów. Zasadniczą zaletą takiego rozwiązania jest redukcja liczby taktów zegarowych koniecznych do wygenerowania pożądaných par testowych. Układ zaczyna pracę od wyzerowania rejestru MISR po czym przez  $g$  taktów zegarowych rejestr MISR pracuje ze słowem programującym  $C_i$ . Po  $g$  taktach zegarowych następuje zmiana stanu licznika adresującego pamięć ROM oraz następuje dostarczenie kolejnego słowa programującego do rejestru MISR i jego ponowne wyzerowanie. Cały proces powtarza się aż do wykorzystania wszystkich  $L_s$  niezbędnych słów programujących. Czas testowania można więc z grubsza określić jako iloczyn  $g \cdot L_s$ . Oczywiście słowa programujące muszą być dobrane w taki sposób, aby podczas  $g$  taktów sygnału zegarowego począwszy od stanu zerowego zostały wygenerowane przez rejestr MISR wszystkie pary testowe związane z danym słowem programującym.

Wadą rozwiązań oferowanych w [2,3] jest duża liczba  $L_s$  słów programujących przechowywanych w pamięci ROM oraz wynikająca z tego zbyt duża liczba taktów zegarowych  $g \cdot L_s$  potrzebnych do realizacji testowania. Inną wadą jest mało elastyczny graf pracy generatora par testowych uniemożliwiający poszukiwanie lepszych rozwiązań.

Podstawowym celem tej pracy jest opracowanie sposobu redukcji liczby  $L_s$  słów programujących tylko do jednego słowa co pozwoli wyeliminować pamięć ROM ze struktury generatora par testowych.

## 2. Rejestr MISR o podwójnej długości

W celu wygenerowania wszystkich możliwych par testowych za pomocą rejestru MISR pracującego tylko z jednym słowem programującym należy podwoić jego długość. Na rysunku (rys. 4.) pokazano rejestr MISR o podwójnej długości z zewnętrznym sprzężeniem zwrotnym. Słowa programujące podawane są na wejścia  $SP = \{s_0, s_1, s_2, \dots, s_{2n-1}\}$ . Jedynka logiczna występująca na pozycji bitu  $s_i$  odpowiada bramce negacji między wyjściem przerzutnika danej sekcji a wejściem przerzutnika kolejnej sekcji. W celu generacji wszystkich  $n$ -bitowych par testowych należy wykorzystać co drugie wyjście podwójnego rejestru MISR  $W = \{w_1, w_3, w_5, \dots, w_{2n-1}\}$ . Oczywiście rejestr MISR musi pracować w oparciu o wybrany wielomian pierwotny  $p_c(x)$ . Aby zastosować konkretny wielomian pierwotny  $p_c(x)$  należy dokonać odpowiednich połączeń wielowejszciowej bramki XOR. Użyty wielomian pierwotny  $p_c(x)$  musi posiadać stopień dwukrotnie większy ( $2n$ ) niż długość par testowych ( $n$ ).



Rys. 4. Struktura rejestru MISR o podwójnej długości  
Fig. 4. Structure of double MISR register

Jak już wcześniej wspomniano każda para testowa  $P = \{V_1, V_2\}$  składa się z dwóch wektorów  $V_1$  i  $V_2$ . Wektor  $V_1$  pobudza uszkodzenie, natomiast wektor  $V_2$  powoduje przesłanie informacji o uszkodzeniu na wyjście testowanego układu. Liczba  $L_{wp}$  wszystkich  $n$ -bitowych par testowych wynosi:

$$L_{wp} = \frac{2^n!}{(2^n - 2)!} \quad (1)$$

Po uproszczeniu uzyskuje się:

$$L_{wp} = 2^n \cdot (2^n - 1). \quad (2)$$

Pełny cykl pracy podwójnego rejestru MISR ( $2n$ ) pracującego w oparciu o wielomian pierwotny  $p_c(x)$  z jednym słowem programującym wynosi:

$$L_{MISR} = 2^{2n} - 1. \quad (3)$$

Z (2) i (3) wynika:

$$L_{MISR} > L_{wp}. \quad (4)$$

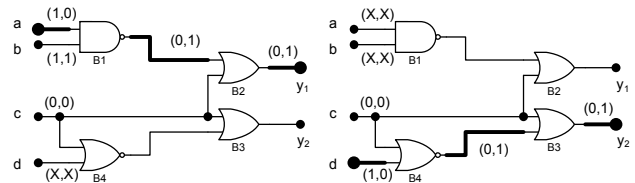
Z (4) wynika, że rejestr MISR o podwójnej długości ( $2n$ ) zbudowany w oparciu o wielomian pierwotny i pracujący tylko z jednym dowolnym słowem programującym wygeneruje wszystkie możliwe  $n$ -bitowe pary testowe. W praktyce okazuje się, że liczba  $L_{pp}$  potrzebnych par testowych wynosi:

$$L_{pp} < L_{wp}. \quad (5)$$

Z (5) wynika, że liczba potrzebnych taktów zegarowych jest mniejsza od liczby wszystkich taktów zegarowych, co jest zasadniczą wadą tego rozwiązania. Jednym ze sposobów zmiany tej sytuacji jest zastosowanie sklejania par testowych [4, 7] co w doprowadzi do zmniejszenia liczby taktów zegarowych.

## 3. Sklejanie par testowych

Na rysunku (rys. 5.) przedstawiono układ kombinacyjny.



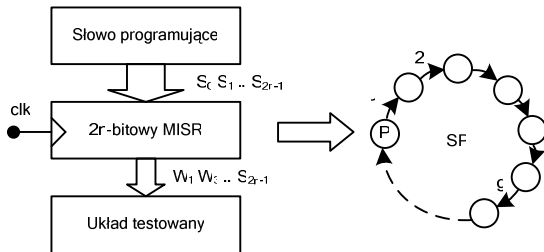
Rys. 5. Prosty układ kombinacyjny  
Fig. 5. Simple structure of combinational circuit

W układzie tym zaznaczono pogrubioną linią dwie ścieżki, które należy poddać testom na występowanie uszkodzeń opóźnionych narastającego zbocza na wyjściach  $y_1$  i  $y_2$ . W tym celu należy wyznaczyć dwie pary testowe po jednej dla każdej ze ścieżek. W przypadku ścieżki a-B<sub>1</sub>-B<sub>2</sub>-y<sub>1</sub> wektor wejściowy  $V_1$  wymuszający stan niski na wyjściu  $y_1$  będzie miał postać:  $V_1 = \{110X\}$ , natomiast wektor wejściowy  $V_2$  wymuszający stan wysoki na wyjściu  $y_1$  będzie miał postać:  $V_2 = \{010X\}$ . Wektory  $V_1$  oraz  $V_2$  tworzą, zatem parę testową  $P_1 = \{110X, 010X\}$  (symbol X oznacza stan nieokreślony). W przypadku ścieżki d-B<sub>4</sub>-B<sub>3</sub>-y<sub>2</sub> wektor wejściowy  $V_1$  wymuszający stan niski na wyjściu  $y_2$  będzie miał postać:  $V_1 = \{XX01\}$ , natomiast wektor wejściowy  $V_2$  wymuszający stan wysoki na wyjściu  $y_2$  będzie miał postać:  $V_2 = \{XX00\}$ . Wektory  $V_1$  oraz  $V_2$  tworzą, zatem parę testową  $P_2 = \{XX01, XX00\}$ . Pary testowe  $P_1$  oraz  $P_2$  są wzajemnie zgodne, więc można je ze sobą skleić w jedną parę testową. Sklejona para testowa będzie miała postać:  $P_{1,2} = \{1101, 0100\}$ .

Z przedstawionego przykładu wynika, że sklejona para testowa  $P_{1,2}$  wystarczy do jednoczesnego przetestowania dwóch ścieżek układu kombinacyjnego. Do tego potrzeba zaledwie dwóch taktów zegarowych. W przypadku nie sklejania par testowych  $P_1$  i  $P_2$  należałoby użyć aż czterech taktów zegarowych do przetestowania tych dwóch ścieżek. W celu generacji przez rejestr MISR o podwójnej długości pary  $P_{1,2} = \{1101, 0100\}$ , należy uzupełnić co drugi bit tej pary o stany nieokreślone ( $P_{1,2} = \{X1X1X0X1, X0X1X0X0\}$ ) po czym wyznaczyć słowo programujące. Stosując algorytmy sklejania par testowych przedstawione w [4, 7] można w znaczący sposób zredukować liczbę potrzebnych par testowych, a tym samym zmniejszyć liczbę potrzebnych taktów zegarowych.

#### 4. Bezpamięciowy generator par testowych

W oparciu o rejestr MISR o podwójnej długości ( $2n$ ) można zrealizować generator par testowych (rys. 6.). Taki generator nie potrzebuje jakiegokolwiek pamięci do przechowywania słów programujących co jest jego główną zaletą.



Rys. 6. Struktura bezpamięciowego generatora par testowych  
Fig. 6. Structure of Test Pattern Generator without ROM

Liczbę taktów zegarowych ograniczono do  $g$  taktów, przy czym:

$$g < 2^{2n} - 1. \quad (6)$$

Struktura generatora par testowych posiada trzy parametry, od których zależy pokrycie par testowych. Należą do nich: stan początkowy  $P$ , słowo programujące  $SP$  oraz liczba taktów zegarowych  $g$ , podczas których pracuje rejestr MISR począwszy od stanu początkowego  $P$ .

#### 5. Wyniki dla układów testowych ISCAS'89

W tabelach (tab. 1. i tab. 2.) przedstawiono wyniki generacji par testowych dla układów testowych ISCAS'89 zaprezentowanych na konferencji ISCAS (International Symposium on Circuits and Systems) w 1989 roku. W pierwszej kolejności sklejono pary testowe za pomocą algorytmu przedstawionego w [4]. Następnie dokonano kilkuset losowań stanu początkowego  $S$  oraz słowa programującego  $SP$  dla założonej liczby taktów zegarowych  $g$ . W celu porównania uzyskanych wyników do [4, 7] użyto podobnej liczby taktów zegarowych  $L_1$ . W pierwszej kolumnie jest nazwa układu testowego (Naz.). W drugiej kolumnie liczba wejść (#we). W trzeciej kolumnie liczba par testowych (#par).

Tab. 1. Wyniki dla układów testowych ISCAS'89  
Tab. 1. Results for ISCAS'89 benchmarks

Naz.	#we	#par	g	$L_s$		$L_1$		Pokrycie [%]	
				TR	[4, 7]	TR	[4, 7]	TR	[4, 7]
c17	5	22	17	1	4	160	68	95,4	95,4
			160						
s27	7	32	32	1	6	256	160	93,7	100
			64						
			256						
s386	13	232	2048	1	49	100000	100352	83,2	93,1
			100000						
s1488	14	738	512	1	89	131072	45568	83,5	71,1
			1024						
			2048						
			131072						
			180000						
s1494	14	725	512	1	94	131072	50176	82,0	70,7
			1024						
			2048						
			131072						
			180000						
s298	17	177	64	1	23	4096	1472	91,5	77,4
			128						
			256						
			4096						
			5400						
s208	18	209	1024	1	28	65536	28672	65,1	65,1
			2048						
			65536						
			131072						

Tab. 2. Wyniki dla układów testowych ISCAS'89  
Tab. 2. Results for ISCAS'89 benchmarks

s832	23	488	1024	1	69	131072	70656	71,1	65,2
			4096						
			262144						
s820	23	475	4096	1	72	131072	294912	71,6	85,5
			131072						
			262144						
s526	24	356	512	1	44	65536	22528	77,5	72,5
			4096						
			65536						
s444	24	303	4096	1	37	131072	151552	74,9	79,5
			131072						
			150000						

W czwartej kolumnie liczba taktów zegarowych, podczas których pracuje rejestr MISR z każdym słowem programującym ( $g$ ). W kolejnej kolumnie jest liczba słów programujących ( $L_s$ ). W następnej kolumnie liczba potrzebnych taktów zegarowych ( $L_1$ ), a w ostatniej kolumnie pokrycie par testowych.

Za pomocą symbolu TR oznaczono wyniki uzyskane za pomocą generatora par testowych zbudowanego w oparciu o rejestr MISR o podwójnej długości. Jako [4, 7] oznaczono wyniki uzyskane za pomocą generatora par testowych z pamięcią ROM (rys. 1.).

#### 6. Wnioski

Jak można zauważyć wyniki jakie uzyskano za pomocą generatora par testowych zbudowanego w oparciu o rejestr MISR o podwójnej długości (rys. 3.) są porównywalne z wynikami jakie uzyskano dla generatora par testowych z pamięcią ROM (rys. 1.). Główną wadą generatora par testowych z pamięcią ROM jest nadmiar sprzętu użytego do jego budowy. Chodzi tutaj głównie o pamięć ROM służącą do przechowywania wielu słów programujących, licznik oraz o dzielnik częstotliwości. Generator par testowych zbudowany w oparciu o rejestr MISR o podwójnej długości nie posiada tej wady, ponieważ do jego pracy wystarcza zaledwie jedno słowo programujące. Jedyną wadą tego rozwiązania jest użycie rejestru MISR o podwójnej długości, który i tak stanowi mniejsze nakłady sprzętowe niż wielokomórkowa pamięć ROM.

#### 7. Literatura

- [1] K. Furuya, E.J. McCluskey: Two-Pattern Test Capabilities of Autonomous TPG Circuits, International Test Conference, 1991.
- [2] M. Keim, I. Polian, H. Hengster, B. Becker: A scalable BIST architecture for delay faults, European Test Workshop, 1999.
- [3] I. Polian, B. Becker: Configuring MISR-Based Two-Pattern BIST Using Boolean Satisfiability, DDECS, Poznań 2003.
- [4] T. Rudnicki, A. Hławiczka: Test Pattern Generator for Delay Faults. DDECS2007, Kraków 2007.
- [5] T. Rudnicki: Two-Pattern Generator With Reduced ROM For Detection Of Delay Faults. MIXDES2005, Kraków 2005.
- [6] A. Hławiczka: Rejestry liniowe - analiza, synteza i zastosowania w testowaniu układów cyfrowych. Zeszyty Naukowe Politechniki Śląskiej Nr 1370, Elektronika, Zeszyt 9, Gliwice 1997.
- [7] T. Rudnicki, A. Hławiczka: Test Pattern Generator for Delay Faults, Theoretical and Applied Informatics, ISSN 1896-5334, Vol. 19, No. 1, 2007.
- [8] J. Savir: Generator Choices for Delay Test, Fourth Asian Test Symposium, 1995.
- [9] A. Efthymiou: Redundancy and Test-Pattern Generation for Asynchronous Quasi-Delay-Insensitive Combinational Circuits, DDECS, Kraków 2007.
- [10] E. Flanigan, A. Abdulrahman, S. Tragoudas: Sequential Path Delay Fault Identification Using Encoded Delay Propagation Signatures, ISQED, 2008.