

Tomasz RUDNICKI, Tomasz GARBOLINO, Krzysztof GUCWA, Andrzej HŁAWICZKA
POLITECHNIKA ŚLĄSKA, INSTYTUT ELEKTRONIKI

Skuteczny generator testów dla przesłuchów w połączeniach

Dr inż. Tomasz RUDNICKI

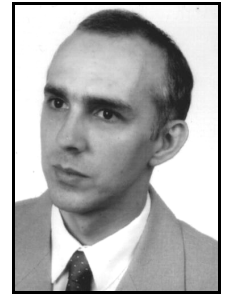
W 2001 roku ukończył studia magisterskie na Wydziale Automatyki, Elektroniki i Informatyki Politechniki Śląskiej na kierunku Elektronika i Telekomunikacja. W 2006 roku obronił pracę doktorską. Od 2006 roku adiunkt w Instytucie Elektroniki Politechniki Śląskiej. Jego zainteresowania naukowe dotyczą testowania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



e-mail: tomasz.rudnicki@polsl.pl

Dr inż. Krzysztof GUCWA

Ukończył studia na Wydziale Automatyki, Elektroniki i Informatyki w 1991 roku. Na tymże wydziale obronił pracę doktorską w 2001 roku. Obecnie jest adiunktem w Instytucie Elektroniki Politechniki Śląskiej. Jego zainteresowania naukowe koncentrują wokół projektowania i testowania układów cyfrowych.



e-mail: krzysztof.gucwa@polsl.pl

Dr inż. Tomasz GARBOLINO

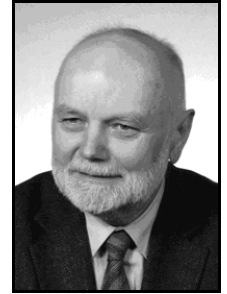
Ukończył studia na Wydziale Automatyki, Elektroniki i Informatyki Politechniki Śląskiej w Gliwicach w 1993 roku. Tam też w 2002 roku obronił pracę doktorską. Obecnie zatrudniony jest jako adiunkt w Instytucie Elektroniki Politechniki Śląskiej. Jego główne zainteresowania naukowe dotyczą zagadnień związanych z testowaniem oraz projektowaniem układów i systemów cyfrowych. Jest współautorem około 40-stu prac z tej dziedziny opublikowanych w czasopiśmie krajowych i zagranicznych.



e-mail: tomasz.garbolino@polsl.pl

Dr hab. inż. Andrzej HŁAWICZKA

Ukończył studia na Wydziale Elektrycznym Politechniki Śląskiej. Pracę doktorską obronił w 1968 r., habilitacyjną w 1998 r. Pracuje w Politechnice Śląskiej w Instytucie Elektroniki od prawie 30 lat. Jest profesorem tej uczelni. W zakresie pracy naukowej zajmuje się konsekwentnie rozwojem nowoczesnych metod diagnostyki i testowania złożonych układów i systemów cyfrowych, w szczególności integrowanych na jednej strukturze krzemowej. Wypromował 3 doktorów nauk technicznych.



e-mail: andrzej.hlawiczka@polsl.pl

Streszczenie

W pracy zasygnalizowano konieczność testowania przesłuchów metodą test-per-clock przy pełnej szybkości zegara w sieciach długich połączeń między modułami w jednocukładowych systemach typu SoC. Do generacji testów zaproponowano rejestr LFSR (ang. Linear Feedback Shift Register) z wielomianem pierwotnym oraz z podwojoną liczbą przerzutników, w którym tylko co drugi przerzutnik jest podłączony do testowanej sieci połączeń. Przeprowadzono eksperymenty symulacyjne sprawdzające skuteczność ich wykorzystania do testowania przesłuchów objawiających się albo chwilowym zakłóceniem (szpilka) albo opóźnieniem zbrocza.

Słowa kluczowe: rejestr liniowy, generator testów, przesłuchy, samotestowanie, system jednocukładowy, sieć połączeń.

Effective BIST for Crosstalk Faults in Inter-connects

Abstract

The paper is devoted to a test-per-clock method of an at-speed testing of crosstalk faults in long interconnects between cores in a System-on-a-Chip. A LFSR composed of $2n$ flip-flops and implementing primitive polynomial was used as a Test Pattern Generator (TPG) for an interconnect network comprised of n nets. In our approach every second output of the LFSR is connected to the Interconnect Network Under Test. Simulation-based experiments were carried out to verify effectiveness of vector sequences produced by the proposed TPG in detection of crosstalk faults provoked at victim net by simultaneous occurrence of rising (falling) edges $01(10)$ at k aggressor lines. Crosstalk faults causing occurrence of a positive (negative) glitch at a victim line having constant value $00(11)$ as well as ones that lead to delaying an edge with an opposite direction $10(01)$ at a victim line were taken into consideration. The experimental results show that for $n \in \{8, 12, 16, 20, 24, 28, 32\}$ and $k \ll n$ all above-mentioned crosstalk faults can be detected by a test sequence having an acceptable length.

Keywords: Test Pattern Generator, Crosstalks, BIST, LFSR, SoC, Interconnect net.

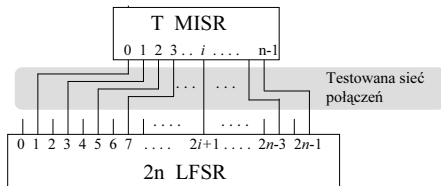
1. Wstęp

Postęp w rozwoju technologii doprowadził do sytuacji, w której cały system może zostać umieszczony w pojedynczym układzie scalonym. Taki system jednocukładowy (ang. System on a Chip – SoC)

ma postać zespołu wbudowanych modułów (ang. core) komunikujących się za pośrednictwem sieci długich połączeń. Wykorzystanie technologii submikronowych (ang. Deep Sub Micron technology – DSM) w produkcji systemów jednocukładowych powoduje zwiększenie pojemności pasożytniczych między połączeniami sąsiednimi geometrycznie. To z kolei prowadzi do powstawania na tyle silnych przesłuchów między tymi połączeniami, że mogą one prowadzić do nieprawidłowego działania systemu lub przynajmniej pogorszenia parametrów jego pracy (np. obniżenia maksymalnej częstotliwości pracy systemu) [2]. Przesłuchy stanowią szczególny problem w przypadku długich połączeń pracujących z bardzo wysokimi częstotliwościami rzędu gigaherców [1, 3]. Wobec tego konieczne jest testowanie tych połączeń z ich nominalną częstotliwością pracy, gdyż przy niższych częstotliwościach zjawisko przesłuchów może się nie ujawnić. Uszkodzenia dynamiczne do wykrycia wymagają trybu "test-per-clock", w którym podawanie pobudzeń testowych przeprowadza się 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 narzędzi wbudowanego samotestowania BIST (ang. Built-In Self-Test) [7, 8, 9]. Przykładem prac poświęconych w ostatnich latach wbudowanemu samotestowaniu typu BIST są prace Jutmana [5]. Jutman przy użyciu BIST typu "test per clock" generuje "Interleaved True /Complement code Sequence". Innym przykładem BIST jest generator testów typu LFSR podłączony do wejść sieci połączeń oraz kompaktor MISR (ang. Multi-Input Signature Register) zbierający odpowiedzi na wyjściach tych połączeń. Rejestry LFSR oraz MISR, zbudowane w tym przypadku na bazie komórek brzegowych, gwarantują testowanie z pełną szybkością zegara [6]. Rejestr MISR, w szczególności ostatnio badany rejestr T-MISR złożony z przerzutników T [10-14], jest bardzo dobrym narzędziem do wykrywania, lokalizacji i identyfikacji uszkodzeń statycznych i dynamicznych w połączeniach. Niestety technika wbudowanego testowania połączeń oparta o generator wektorów testowych w postaci rejestru LFSR ma kilka wad. Najistotniejszą wadą tej techniki jest brak możliwości wygenerowania niektórych par testów koniecznych dla pobudzenia konkretnych przesłuchów. W klasycznym rejestrze przesuwającym po ustawieniu pary jego stopni i oraz $i+1$ parą stanów xy jest niemożliwe wygenerowanie na tych stopniach w następnym taktie zegarowym pary stanów zx' gdzie x' jest zanegowanym stanem x . Jest to

wynikiem równości stanu następnego stopnia $i+1$ oraz stanu poprzedniego stopnia i -tego w trakcie przesuwania bitów w rejestrze przesuwającym. Jeżeli stan poprzedni stopnia i -tego miał wartość x to stan następnego stopnia $i+1$ także będzie miał wartość x .

Zakładając, że generatorem testów jest LFSR wówczas sprzętowa procedura generacji testów wymusza przesunięcie o jeden bit zawartości rejestru zanim kolejny wektor zostanie podany na wejścia TUC. Na rys. 1 przedstawiono generator, który ma podwojoną liczbę przerzutników $2n$ w stosunku do liczby normalnie potrzebnej do wygenerowania n -bitowych par testowych.



Rys. 1. BIST dla sieci n -połączeń
Fig. 1. BIST for net composed of n interconnects

Dalej będzie on oznaczany jako rejestr $2n$ LFSR. Do testowanej sieci połączeń jest podłączony tylko ko drugi przerzutnik. Taki sposób podłączenia eliminuje tzw. „shift dependency”. Ponieważ stany przerzutników parzystych są niezależne od zawartości przerzutników nieparzystych dlatego taki generator wygeneruje wszystkie możliwe pary testów. Cykl pracy rejestru wynosi:

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

Po raz pierwszy ideę takiego generatora par testowych dla pobudzenia uszkodzeń opóźnieniowych podał Savir w [4]. Celem naszej pracy jest wykazanie, że tak skonstruowany generator testów z podwojoną liczbą przerzutników $2n$ jest skutecznym generatorem n -bitowych par testowych dla pobudzenia określonych przesłuchów w sieci połączeń.

2. Motywacja

W najprostszym przypadku w wyniku przesłuchu stan jednej linii nazywanej agresorem wpływa na stan innej linii v nazywanej ofiarą. Powyższy wpływ może się objawiać w dwojaki sposób. Z jednej strony pojawienie się zbocza 01 (10) na linii agresora może spowodować powstanie chwilowego zakłócenia – tzw. szpilki (ang. glitch) – na linii ofiary o stałej wartości $v = 00$ ($v = 11$). Z drugiej strony wystąpienie zbocza o jednym kierunku np. 10 (01) na linii agresora może doprowadzić do opóźnienia pojawienia się na linii ofiary v zbocza o przeciwnym kierunku 01 (10). W dalszej części artykułu oba zjawiska będą nazywane przesłuchami typu odpowiednio G (ang. Glitch) i D (ang. Delay). W bardziej złożonych przypadkach do wystąpienia przesłuchu konieczne jest jednocześnie pojawienie się zbocza o tym samym kierunku na kilku liniach agresorów. W celu określenia liczby k agresorów powodujących przesłuch typu D lub G przesłuchy te oznaczać będziemy odpowiednio przez D_k oraz G_k . Mechanizm powstawania przesłuchów determinuje rodzaj testów zapewniających ich pobudzenie. Ponieważ do powstania przesłuchu konieczne jest wystąpienie zbocza na liniach agresorów więc niezbędne jest podanie na sieć połączeń pary wektorów testowych. Wybrane pary wektorów pobudzające omówione przesłuchy podano w tab. 1.

Tab. 1. Wybrane pary wektorów testowych
Tab. 1. Examples of Two-Test Patterns

1	2	3	4
$G_1; v = 00$	XX...X00X...XXXX XX...X10X...XXXX	$G_1; v = 11$	XX...X11X...XXXX XX...X01X...XXXX
$D_1; v = 10$	XX...X01X...XXXX XX...X10X...XXXX	$D_1; v = 01$	XX...X10X...XXXX XX...X01X...XXXX
$G_3; v = 00$	XX...X000X...XX XX...X1011X...XX	$G_3; v = 11$	XX...X1111X...XX XX...X0100X...XX
$D_3; v = 10$	XX...X0100X...XX XX...X1011X...XX	$D_3; v = 01$	XX...X1011X...XX XX...X0100X...XX

W kolumnach 1 i 3 podano typ przesłuchu, natomiast w kolumnach 2 i 4 pary wektorów pobudzających dany przesłuch. W niniejszej pracy założono, że topologia n połączeń nie jest znana. Tego typu założenie wymaga wygenerowania wszystkich możliwych par pobudzających wszystkie cztery typy przesłuchów $\{D_k; v = 01\}$, $\{D_k; v = 10\}$, $\{G_k; v = 00\}$, $\{G_k; v = 11\}$.

Zbiór wszystkich możliwych par testów dla k agresorów oraz sieci n połączeń oznaczany będzie w pracy przez $S_{k,n}$. Liczność takiego zbioru można wyrazić wzorem:

$$L = 4n \binom{n-1}{k}. \quad (2)$$

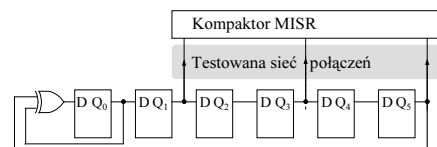
Przykład zbioru $S_{1,3}$ zawierającego wszystkie 24 możliwe pary testowe pobudzające przesłuchy $\{D_1; v = 01\}$, $\{D_1; v = 10\}$, $\{G_1; v = 00\}$, $\{G_1; v = 11\}$ w sieci 3 połączeń jest pokazany w tab. 2.

Tab. 2. Zbiór $S_{1,3}$ par testowych
Tab. 2. Two-Test Pattern set $S_{1,3}$

D_1	01X 10X	0X1 1X0	X01 X10	10X 01X	1X0 0X1	X10 X01
G_1 $v = 00$	00X 10X	0X0 1X0	X00 X10	00X 01X	0X0 0X1	X00 X01
G_1 $v = 11$	11X 10X	1X1 1X0	X11 X10	11X 01X	1X1 0X1	X11 X01

3. Generator par testowych $2n$ LFSR

Rozpatrzmy przykład sieci $n = 3$ połączeń oraz przykład rejestru $2n$ LFSR o wielomianie pierwotnym $p(x) = 1 + x^5 + x^6$ zawierającym 6 przerzutników Q_0, Q_1, Q_2, Q_3, Q_4 oraz Q_5 , z których przerzutniki Q_1, Q_3 oraz Q_5 są podłączone do wejść sieci połączeń. Schemat generatora testów przedstawiono na rys. 2.



Rys. 2. Schemat generatora par testowych z wielomianem $p(x) = 1 + x^5 + x^6$
Fig. 2. Scheme of Two-Test Pattern Generator with polynomial $p(x) = 1 + x^5 + x^6$

W tabeli 3 przedstawiono kolejne wektory ciągu T generowanego przez rejestr $2n$ LFSR zakładając, że został on wstępnie ustawiony w stan 101000. W tabeli tej jednocześnie zaznaczono tustymi czcionkami wszystkie te zbocza, które pobudzają przesłuchy typu D_1 , a więc te, które zawierają zbocza o jednym kierunku na linii agresora oraz zbocza o przeciwnym kierunku na linii ofiary.

Tab. 3. Fragment sekwencji T pobudzającej przesłuchy D_1
Tab. 3. Part of T sequence stimulating D_1 type crosstalks

	Q_0	Q_1^*	Q_2	Q_3^*	Q_4	Q_5^*		Q_0	Q_1^*	Q_2	Q_3^*	Q_4	Q_5^*
00	1	0	1	0	0	0	10	0	0	1	0	1	0
01	1	1	0	1	0	0	11	0	0	0	1	0	1
02	1	1	1	0	1	0	12	1	0	0	0	1	0
03	1	1	1	1	0	1	13	1	1	0	0	0	1
04	0	1	1	1	1	0	14	0	1	1	0	0	0
05	0	0	1	1	1	1	15	0	0	1	1	0	0
06	1	0	0	1	1	1	16	0	0	0	1	1	0
07	0	1	0	0	1	1	17	0	0	0	0	1	1
08	1	0	1	0	0	1	18	1	0	0	0	0	1
09	0	1	0	1	0	0	19	0	1	0	0	0	0

Pary testów ciągu T pobudzające wszystkie przesłuchy typu D_1 na każdej z trzech linii Q_1, Q_3 , oraz Q_5 występują w następujących taktach zegarowych: $\langle 4,5 \rangle$, $\langle 6,7 \rangle$, $\langle 8,9 \rangle$, $\langle 14,15 \rangle$, $\langle 16,17 \rangle$ oraz $\langle 18,19 \rangle$. W tab. 4 zaznaczono z kolei tustymi cyframi zbocza na tych pojedynczych liniach agresorów, które mogą spowodować powstanie przesłuchu typu G_1 - na linii ofiary o stałej wartości 1. Pary testów ciągu T pobudzających takie przesłuchy związane są z następującymi taktami zegarowymi: $\langle 1,2 \rangle$, $\langle 3,4 \rangle$, $\langle 4,5 \rangle$, $\langle 6,7 \rangle$, $\langle 7,8 \rangle$.

Tab. 4. Fragment sekwencji T pobudzającej przesłuchy G_1
 Tab. 4. Part of T sequence stimulating G_1 type crosstalks

	Q_0	Q_1^*	Q_2	Q_3^*	Q_4	Q_5^*		Q_0	Q_1^*	Q_2	Q_3^*	Q_4	Q_5^*
00	1	0	1	0	0	0	10	0	0	1	0	1	0
01	1	1	0	1	0	0	11	0	0	0	1	0	1
02	1	1	1	0	1	0	12	1	0	0	0	1	0
03	1	1	1	1	0	1	13	1	1	0	0	0	1
04	0	1	1	1	1	0	14	0	1	1	0	0	0
05	0	0	1	1	1	1	15	0	0	1	1	0	0
06	1	0	0	1	1	1	16	0	0	0	1	1	0
07	0	1	0	0	1	1	17	0	0	0	0	1	1
08	1	0	1	0	0	1	18	1	0	0	0	0	1
09	0	1	0	1	0	0	19	0	1	0	0	0	0

Pary testów ciągu T pobudzających przesłuchy typu G_1 na liniach o stałych wartościach 0 opisane są następującym zbiorem par: $\langle 0,1 \rangle$, $\langle 10,11 \rangle$, $\langle 12,13 \rangle$, $\langle 14,15 \rangle$, $\langle 16,17 \rangle$, $\langle 18,19 \rangle$ w tabeli 4.

Zauważmy, że ustawiając rejestr LFSR w stan początkowy 101000 otrzymujemy w ciągu testowym T w jego kolejnych 19 taktach zegarowych wszystkie, zawarte w zbiorze $S_{1,3}$, pary testów pobudzających przesłuchy typu D_1 oraz G_1 na każdej z trzech linii Q_1 , Q_3 , oraz Q_5 . Zatem nie jest potrzebny cały cykl $2^{2n} - 1$ czyli 63 stanów do wygenerowania wszystkich niezbędnych par testowych. Zwróćmy uwagę również na fakt, że liczba 19 taktów zegarowych jest również mniejsza od liczności $L = 24$ zbioru $S_{1,3}$ zawierającego wszystkie możliwe pary testowe.

Przykład ten umotywowwał autorów do przeprowadzenia eksperymentów sprawdzających skuteczność testowania za pomocą rejestrów $2n$ LFSR przesłuchów D_k oraz G_k dla magistral zawierających 8 i więcej połączeń oraz $k \ll n$.

4. Eksperymenty

Oznaczmy przez g najmniejszą długość takiego ciągu T, który zawiera wszystkie pary testów niezbędnych do pobudzenia założonych przesłuchów typu D_k oraz G_k na założonej magistrali n połączeń. Celem eksperymentów było określenie wartości g , znacząco mniejszej od liczności L zbioru $S_{k,n}$. Eksperymenty przeprowadzono dla magistral zawierających 8, 12, 16, 20, 24, 28 oraz 32 połączenia. Założono jednocześnie, że generowane na tych magistralach sekwencje T zawierają n -bitowe pary testowe pobudzające przesłuchy ze zbioru $P = \{D_1, D_2, D_3, D_4, G_1, G_2, G_3, G_4\}$ co spełnia warunek $k \ll n$. W celu przeprowadzenia eksperymentu najpierw wygenerowano dla każdej z n -bitowych magistral oraz dla k agresorów wszystkie zbiory $S_{k,n}$ zawierające wszystkie możliwe pary testów pobudzające przesłuchy ze zbioru P. Następnie w przypadku każdej magistrali i każdej liczby k agresorów 100 razy losowano stan początkowy i wybierano ten, który gwarantował, że ciąg T zawierał 100% wszystkich określonych wcześniej par testowych ze zbioru $S_{k,n}$, a jego długość g była najmniejsza i jednocześnie znacząco mniejsza od L . Eksperymenty symulacyjne przeprowadzono przy użyciu jednego komputera Pentium P4 (3,4 GHz). Rezultaty eksperymentów przedstawiono w tab. 5.

Tab. 5. Rezultaty eksperymentów dla przesłuchów D_3, G_3, D_4, G_4
 Tab. 5. Experimental results for crosstalks D_3, G_3, D_4, G_4

$p_n(x)$	k	n	L	g	t
$p_8(x) = 1+x^2+x^3+x^5+x^{16}$	3	8	1120	986	1s
$p_{12}(x) = 1+x^2+x^3+x^4+x^{24}$	3	12	7920	1406	12s
$p_{16}(x) = 1+x^2+x^6+x^7+x^{32}$	3	16	29120	1410	45s
$p_{20}(x) = 1+x^3+x^4+x^5+x^{40}$	3	20	77520	1840	2min 2s
$p_{24}(x) = 1+x^4+x^7+x^9+x^{48}$	3	24	170016	2099	5min 7s
$p_{28}(x) = 1+x^2+x^4+x^7+x^{56}$	3	28	327600	2192	10min
$p_{32}(x) = 1+x^3+x^4+x^{64}$	3	32	575360	2338	47min
$p_8(x) = 1+x^2+x^3+x^5+x^{16}$	4	8	1120	4478	7s
$p_{12}(x) = 1+x^2+x^3+x^4+x^{24}$	4	12	15840	6454	2min
$p_{16}(x) = 1+x^2+x^6+x^7+x^{32}$	4	16	87360	8177	20min
$p_{20}(x) = 1+x^3+x^4+x^5+x^{40}$	4	20	310080	9191	35min
$p_{24}(x) = 1+x^4+x^7+x^9+x^{48}$	4	24	850080	10038	1,5h
$p_{28}(x) = 1+x^2+x^4+x^7+x^{56}$	4	28	1965600	11378	3,5h
$p_{32}(x) = 1+x^3+x^4+x^{64}$	4	32	4027520	12466	10,5h

Ilustrują one wyniki generacji par testowych pobudzających przesłuchy odpowiednio ze zbiorów $\{D_3, G_3\}$, $\{D_4, G_4\}$. Czasy symulacji t umieszczono w prawej skrajnej kolumnie. Uzyskane czasy symulacji generacji par testowych dla wszystkich stu stanów początkowych (ang. seed) są akceptowalne. Czas ten, jak można zauważyć w tabeli 6 wzrasta wraz ze wzrostem liczby par testowych L. Zwróćmy jednocześnie uwagę, że drastyczny wzrost liczności L zbioru $S_{k,n}$ nie wpływa na znaczący wzrost długości g sekwencji T. Zatem jest nadzieja, że dla magistral o liczbie $32 < n < 64$ oraz dla liczby agresorów $k > 4$ długość g ciągu T będzie jeszcze akceptowalna. Ze względu na to, że losowanie stanów początkowych można równoległe przy użyciu wielu komputerów, realizowalne staje się skrócenie czasu symulacji t dla takich wartości n oraz k .

5. Podsumowanie

W pracy zaproponowano strukturę generatora par testowych pobudzających przesłuchy ze zbioru $P = \{D_1, D_2, D_3, D_4, G_1, G_2, G_3, G_4\}$. Jest nim rejestr $2n$ LFSR z dowolnym wielomianem pierwotnym, który ma podwojoną liczbę przerzutników $2n$ w stosunku do liczby normalnie potrzebnej do wygenerowania n -bitowych par testowych. Do testowanej sieci n połączeń jest podłączony tylko co drugi przerzutnik tego rejestru. Przeprowadzone eksperymenty udowodniły, że dla przesłuchów D_k oraz G_k gdzie $k \ll n$ można w krótkim czasie t znaleźć losowy stan początkowy gwarantujący, że generowany przez rejestr $2n$ LFSR ciąg T o akceptowalnej długości g zawierać będzie 100 % wszystkich par testowych niezbędnych do pobudzenia przesłuchów D_k oraz G_k .

6. Literatura

- [1] K. Shu-Min Li, C. Len Lee, C. Su, J. Chen: A Unified Approach to Detecting Crosstalk Faults of Interconnects in Deep Sub-Micron VLSI. ATS, str. 145-150, 2004.
- [2] X. Aragones, J.L. González, F. Moll, and A. Rubio. Noise Generation and Coupling Mechanisms in Deep-Submicron ICs. IEEE Design & Test Computers, vol.19, no.5, str.27-35, 2002.
- [3] The International Technology Roadmap for Semiconductors, 2007, URL: <http://public.itrs.net/>
- [4] J. Savir: Generator Choices for Delay Test, ATS, pp. 214-221, 1995.
- [5] A. Jutman: At-Speed On-Chip Diagnosis of Board-Level Interconnect Faults. ETS, 2004.
- [6] B.Nadeau-Dostie: An Embedded Technique for At-Speed Interconnect Testing. ITC, str.431-438, 1999.
- [7] R.Pendurkar,A.Chatterjee,Y.Zorian: Switching Activity Generation with Automated BIST Synthesis for Performance Testing of Interconnects. IEEE Trans CAD/ICS, str.20, n.9, 2001.
- [8] A.Attarha, M.Nourani. Testing Interconnects for Noise and Skew in Gigahertz SoC. ITC, str.305-314, 2001.
- [9] C.Su,W.Tseng: Configuration Free SoC Interconnect BIST Methodology. ITC, str.1033-1038, 2001.
- [10] M. Kopec, T. Garbolino, K. Gućwa, A. Hławiczka: Test-per-clock detection, localization and identification of interconnect faults. ETS, str. 233-238, 2006.
- [11] T. Garbolino, M. Kopec, K. Gućwa, A. Hławiczka: Detection, Localisation and Identification of Interconnection Fault Using MISR Compactor. DDECS, str. 230-231, 2006.
- [12] A. Hławiczka, K. Gućwa, T. Garbolino, M. Kopec: Can a D flip-flop based MISR compactor reliably detect interconnect faults?. DDECS, str.2-8, 2005.
- [13] M. Kopec, T. Garbolino, K. Gućwa, A. Hławiczka: On Application of Polynomial Algebra for Identification of Dynamic Faults in Interconnects. Electronics and Telecommunications Quarterly, 54, no 1, str. 29-41, 2008.
- [14] M. Kopec, T. Garbolino, K. Gućwa, A. Hławiczka: Identification of Dynamic Faults in Interconnects by use of Polynomial Algebra. Electronic Symposium Digest of Papers, ETS, 2008.