

Ireneusz MROZEK<sup>1</sup>  
Vyacheslav YARMOLIK<sup>2</sup>

### **WYKORZYSTANIE CHARAKTERYSTYKI ADRESOWEJ W TRANSPARENTNYM TESTOWANIU PAMIĘCI RAM**

*Artykuł przedstawia strategię transparentnego testowania pamięci RAM opartą o charakterystykę adresową. Charakterystyka ta jest otrzymywana jako suma modulo 2 adresów wszystkich komórek pamięci, których zawartość jest równa 1. Charakterystyka ta może zostać wykorzystana jako sygnatura odniesienia w procesie testowania transparentnego. W artykule zostaną zaprezentowane podstawowe własności powyższego podejścia jak również zamieszczone zostaną wyniki otrzymane drogą symulacji potwierdzające wysoką wydajność tak realizowanych testów transparentnych.*

### **TRANSPARENT RAM TESTING BASED ON SELF-ADJUSTING OUTPUT DATA COMPRESSION**

*In this paper the new concept of transparent RAM testing based on Self-Adjusting Output Data Compression (SAODC) have been proposed and developed. The presented technique is based on a memory characteristic derived as the modulo-2 sum of all addresses pointing to non-zero memory cells. This characteristic can be used as the RAM under test fault-free signature. The main properties of new transparent tests are investigated and experimentally validated.*

## **1. WSTĘP**

Złożone pamięci półprzewodnikowe stanowią integralną część większości współczesnych modułów elektronicznych będących składowymi urządzeniami technicznymi wykorzystywanymi w wielu dziedzinach życia. Zjawiskiem codziennym są dziś urządzenia cyfrowe pełniące rolę systemów o znaczeniu krytycznym. Efektem tego jest ciągły wzrost wymagań dotyczących długotrwałej, niezawodnej i często nieprzerwanej pracy, zarówno pojedynczych układów scalonych, pakietów, urządzeń, jak również całych systemów cyfrowych. Z danych statystycznych wynika, że około 70% wszystkich uszkodzeń w systemach cyfrowych spowodowanych jest uszkodzeniami pamięci. Dlatego już na etapie projektowania należy zadbać o możliwość łatwego i efektywnego ich testowania. Jednym z najczęściej stosowanych rozwiązań jest projektowanie układów z myślą o testowaniu –

---

<sup>1</sup>Politechnika Białostocka, Wydział Informatyki, 15-351 Białystok, ul Wiejska 45a, e-mail: i.mrozek@pb.edu.pl

<sup>2</sup>Politechnika Białostocka, Wydział Informatyki, 15-351 Białystok, ul Wiejska 45a, e-mail: v.yarmolik@pb.edu.pl

DFT (ang. design for testability). Z uwagi na coraz większą integrację układów cyfrowych, umieszczaniu wielu modułów (w tym pamięci) na jednym podłożu krzemowym, tradycyjne „zewnętrzne” techniki testowania, nie zawsze mogą być stosowane. Rozwiązaniem problemu okazały się bogato opisane w literaturze techniki testów wbudowanych BIST (ang. built-in-self test) i samonaprawy BISR (ang. built-in-self repair) [1-7].

W procesie wbudowanego testowania pamięci szeroko zaakceptowane zostały testy krokowe (ang. march tests) łączące wysoką wykrywalność uszkodzeń z niską złożonością testu rzędu  $O(N)$  [8]. Ponadto klasyczne testy krokowe są łatwo konwertowalne do testów transparentnych (ang. transparent march tests) [9-12]. Testy transparentne umożliwiają zachowanie niezmięnionej zawartości pamięci w stosunku do zawartości z momentu rozpoczęcia testu. Dzięki temu są one szczególnie predysponowane do realizacji okresowych testów wykonywanych w czasie normalnej pracy urządzenia [8]. Jednak wraz ze wzrostem rozmiarów pamięci czas niezbędny na jej przetestowanie ulega również wydłużeniu. Problem ten jest szczególnie widoczny w wypadku testów transparentnych. Przy klasycznym podejściu każdorazowa realizacja testu wymaga dodatkowych nakładów obliczeniowych niezbędnych do wyliczenia sygnatury odniesienia. Złożoność tego procesu sięga około 50% oryginalnej złożoności realizowanego testu [9].

W artykule przedstawiony zostanie transparentny sposób testowania umożliwiający uniknięcie niekorzystnych narzutów czasowych związanych z wyliczaniem sygnatury odniesienia. Wykorzystana zostanie technika SAODC (ang. Self-Adjusting Output Data Compression) zaproponowana oryginalnie do testów nietransparentnych [13]. Okazuje się, iż wykorzystanie powyższej techniki w testowaniu transparentnym pozwala zminimalizować czas niezbędny do wyliczenia sygnatury odniesienia. Tak realizowane testy cechują się tą samą wydajnością co realizacje standardowe, a przy niewielkich modyfikacjach implementacyjnych ich wydajność nawet wzrasta co zostanie potwierdzone wynikami przeprowadzonych badań.

Artykuł podzielony jest na pięć punktów. Podstawowa idea techniki SAODC przedstawiona zostanie Czytelnikowi w punkcie 2. W punkcie 3 zaprezentowane zostanie wykorzystanie techniki SAODC do realizacji testów transparentnych. Przeanalizowane będą tam również własności testowania realizowanego w oparciu o SAODC. Punkt 4 zawiera analizę efektywności proponowanego podejścia, zaś w punkcie 5 nastąpi podsumowanie przedstawianych treści.

## 2. CHARAKTERYSTYKA ADRESOWA

Idea techniki SAODC testowania pamięci wbudowanej, wykorzystująca charakterystykę adresową pamięci, przedstawiona została szczegółowo w [13]. Charakterystyka odniesienia (sygnatura odniesienia)  $S_R$  początkowego (poprawnego) stanu pamięci definiowana jest jako suma modulo-2 adresów wszystkich komórek przechowujących wartość „1”. Zgodnie z teorią testowania opartego o SAODC podczas testu analizowane są wszystkie komórki pamięci oprócz komórki o adresie 0 [13]. W zapisie bardziej formalnym powyższe podejście można przedstawić w sposób następujący. Oznaczmy pamięć bitową o liczbie komórek równej  $N=2^m-1$  z przestrzenią adresową  $A=\{1,2,3,\dots,2^m-1\}$  przez  $M[a]$ ,  $a \in A$ . Niech  $A_1:=\{a \in A \mid M[a] = 1\}$  oznacza zbiór wszystkich adresów komórek pamięci zawierających wartość „1”, a  $A_0:=\{a \in A \mid M[a] = 0\}$  zbiór wszystkich adresów komórek pamięci zawierających wartość „0”, gdzie  $A_1 \cup A_0 = A$ .

Wtedy charakterystykę odniesienia  $S_R$  [13] otrzymujemy poprzez zsumowanie modulo-2 wszystkich adresów z  $A_1$ .

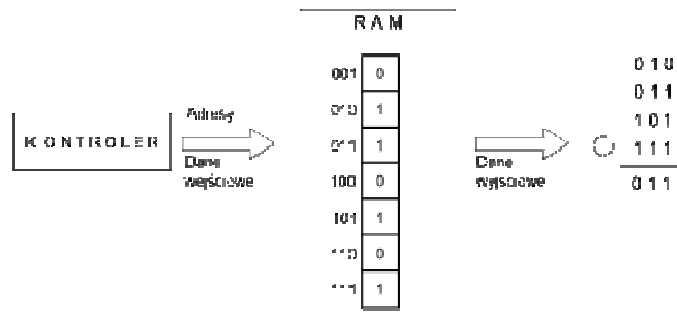
$$S_R = \bigoplus_{a \in A_1} a \tag{1}$$

Dla powyższej charakterystyki prawdziwe są również następujące relacje:

$$S_R = \bigoplus_{a \in A_1} a = S_{R_1} = \bigoplus_{a \in A_1} a = S_{R_0} = \bigoplus_{a \in A_0} a \tag{2}$$

$$S_{RT} = \bigoplus_{a \in A} a = 0^m . \tag{3}$$

W tym wypadku zapis  $0^m$  oznacza sekwencję złożoną z  $m$  powtórzeń wartości 0. Używając powyższego zapisu sekwencję 000110001100011 można zapisać jako  $(0^3 1^2)^3$ . Z (2) wynika, iż dla pamięci o rozmiarze  $N=2^m-1$  charakterystyka w której sumowane są adresy wszystkich komórek z wartościami 1 jest równa charakterystyce w której sumowane są wszystkie adresy komórek z wartościami 0 ( $S_{R1}=S_{R0}$ ). Prosty przykład wyliczania sygnatury odniesienia tą techniką w wypadku  $2^m-1 = 2^3-1=7$  komórek pamięci przedstawia rysunek 1.



Rys.1. Charakterystyka adresowa  $S_R$  bitowej pamięci RAM

W przykładzie z rys. 1  $S_{R1}=010 \oplus 011 \oplus 101 \oplus 111=011$ ,  $S_{R0}=001 \oplus 100 \oplus 110=011$ , zaś suma  $S_{R1}$  i  $S_{R0}$  zgodnie z (3) jest równa 000 ( $S_{R1} \oplus S_{R0}=011 \oplus 011=000$ ). Na podstawie (1) można również zdefiniować następującą własność [13]:

**Własność 1.** W technice testowania pamięci SAODC uszeregowanie adresów podczas wyliczania sygnatury  $S_R$  może być przyjęte w sposób dowolny.

Wynika z niej, iż kolejność adresów sumowanych podczas wyliczania charakterystyki  $S_R$  nie ma znaczenia dla końcowej wartości tej charakterystyki.

### 3. SAODC W TESTOWANIU TRANSPARENTNYM

Kluczową rolę w technikach testowania pamięci odgrywają testy krokowe (testy typu March) określane również w literaturze polskiej jako testy maszerujące [14]. Test typu March składa się ze skończonej liczby elementów typu March. Element typu March (faza testu) składa się ze skończonej liczby operacji, z których wszystkie oddziałują na określoną komórkę przed przejściem do następnej komórki pamięci. Komórka następna określona jest poprzez sposób adresowania, który może być wzrastający, w którym adresy wzrastają od 0 do  $N-1$ , oznaczany przez  $\uparrow$ , lub malejący, w którym adresy maleją od  $N-1$  do 0, oznaczany przez  $\downarrow$ . W niektórych wypadkach sposób adresowania można wybrać dowolnie i jest to oznaczane przez  $\updownarrow$ . Możliwe operacje dostępne na komórkach pamięci dostępne w testach krokowych to:  $wx$  – zapisz do komórki pamięci wartość  $x$  i operacja  $rx$  – odczytaj wartość z komórki pamięci (wartość spodziewana jest równa  $x$ ). Cały test typu March ograniczony jest poprzez parę nawiasów klamrowych { ... }, podczas gdy faza testu krokowego ograniczona jest parą nawiasów okrągłych ( ... ). Jako przykład rozpatrzmy test MATS+ [8]:

$$\{ \updownarrow(w0); \uparrow(r0, w1); \downarrow(r1, w0) \}.$$

$M0 \qquad M1 \qquad M2$

Test ten składa się z trzech faz: M0, M1, M2. Faza M0 zeruje pamięć. Faza M1 w porządku adresów rosnących czyta wartość 0 z komórki, następnie zapisuje tam wartość 1. Faza M2 w porządku adresów malejących odczytuje wartość 1 z komórki, po czym do tej samej komórki wpisuje wartość 0. W tradycyjnych testach krokowych na uszkodzenie pamięci wskazuje fakt odczytania innej wartości niż wartość spodziewana.

Możliwość wystąpienia uszkodzenia w trakcie eksploatacji układu pamięci wymaga cyklicznego sprawdzania jakości jej działania bez niszczenia przechowywanej tam zawartości. Można to osiągnąć używając transparentnej metody testowania. Zgodnie z procedurą, zaproponowaną przez M. Nicloaidis'a [9], każdy tradycyjny test krokowy można przekształcić do postaci transparentnej. Procedura transformacji sprowadza się do następujących kroków:

1. Pominąć w teście fazę inicjacyjną nie mającą wpływu na jakość testu. Jej zadaniem jest jedynie wstępne wypełnienie pamięci określonymi wartościami (zazwyczaj wartością 0).
2. Wszystkie statyczne polecenia odczytu ( $r0$ ,  $r1$ ) i zapisu ( $w0, w1$ ) zastąpić poleceniami uogólnionymi w postaci  $ra$ ,  $r\bar{a}$ ,  $wa$ ,  $w\bar{a}$ . Uogólnione polecenia  $ra$  ( $r\bar{a}$ ) oznaczają odczytanie z pamięci wartości  $a$  ( $\bar{a}$ ) oraz przekazanie ich do analizatora sygnatury.
3. W każdej fazie testu, w której przyjęty jest dowolny sposób adresowania, przyjmując jeden z określonych sposobów adresowania (np. w danej fazie symbol  $\updownarrow$ , oznaczający dowolny sposób adresowania, zastąpić symbolem  $\uparrow$  lub  $\downarrow$ , jednoznacznie opisującym sposób adresowania pamięci).

Po tych trzech krokach otrzymana zostanie bazowa transparentna wersja testu poddanego

transformacji. W celu pełnej realizacji testu niezbędne jest wyznaczenie algorytmu pozwalającego określić sygnaturę odniesienia. Sposób otrzymania tej sygnatury opisany jest w czwartym kroku przekształceń.

4. Z otrzymanego w trzecim kroku testu usunąć wszystkie polecenia zapisu. Polecenia odczytu ( $ra$  i  $r\bar{a}$ ) zaimplementować tak, aby po poleceniu  $ra$  do rejestru sygnatury skierowana została odczytana wartość, zaś po poleceniu  $r\bar{a}$  wartość przeciwna do wartości odczytanej.

W tym wypadku rezultat testu jest wynikiem porównania wartości sygnatury odniesienia (sygnatury pamięci z momentu poprzedzającego rozpoczęcie testu) z wartością sygnatury pamięci wyliczonej po zakończeniu testu (lub wyliczanej w trakcie jego trwania). Zatem w wypadku tradycyjnej postaci testowania transparentnego zaproponowanej w [9], transponowany test jest rozszerzany o fazę wyznaczenia sygnatury odniesienia  $S_{test}$ . Zwiększa to o około 50% czas realizacji testu. Przykładowo test March C- w wersji nietransparentnej ma postać:  $\Downarrow(w0)$ ;  $\Uparrow(r0,w1)$ ;  $\Uparrow(r1,w0)$ ;  $\Downarrow(r0,w1)$ ;  $\Downarrow(r1,w0)$ ;  $\Downarrow(r0)$  i złożoność  $10N$ . Złożoności transparentnej wersji bazowej  $\Uparrow(ra,w\bar{a})$ ;  $\Uparrow(r\bar{a},wa)$ ;  $\Downarrow(ra,w\bar{a})$ ;  $\Downarrow(r\bar{a},wa)$ ;  $\Uparrow(ra)$  równa jest  $9N$ , zaś procedury wyliczania sygnatury odniesienia  $\Uparrow(ra)$ ;  $\Uparrow(r\bar{a})$ ;  $\Downarrow(ra)$ ;  $\Downarrow(r\bar{a})$ ;  $\Uparrow(ra) - 5N$ . Zatem finalna złożoność transparentnej procedury testującej opartej o test March C- przy podejściu tradycyjnym jest równa  $14N$ . Przy wciąż zwiększających się rozmiarach pamięci zwiększająca się złożoność testu jest bardzo niekorzystna.

Podczas realizacji testu krokowego można wyróżnić dwie podstawowe zawartości pamięci RAM: oryginalną  $D=d_1d_2d_3\dots d_{N-1}$  (zawartość pamięci z momentu bezpośrednio poprzedzającego rozpoczęcie testu) oraz zawartość komplementarną w stosunku do zawartości oryginalnej  $D^*=d_1^*d_2^*d_3^*\dots d_{N-1}^*$  składającą się z zanegowanych wartości oryginalnych. Przykładowo, rozważając test March C-:  $\{\Uparrow(ra,w\bar{a});\Uparrow(r\bar{a},wa);\Downarrow(ra,w\bar{a})\Downarrow(r\bar{a},wa)\Uparrow(ra)\}$ , podczas pierwszej, trzeciej i piątej fazy odczytywane są z pamięci wartości oryginalne  $D=d_1d_2d_3\dots d_{N-1}$ , zaś w fazie drugiej i czwartej z pamięci odczytywane są wartości zanegowane  $D^*=d_1^*d_2^*d_3^*\dots d_{N-1}^*$ . W wypadku zastosowania techniki SAODC powyższe spostrzeżenie oraz równanie (2) pozwalają zmniejszyć złożoność wyliczania sygnatury odniesienia i zredukować ją do pojedynczej instrukcji. W tym celu w procedurze konwersji testu krokowego na postać transparentną należy pominąć p. 4 i zmodyfikować p.1 na postać: „Zamienić fazę inicjacyjną transponowanego testu  $\Downarrow(w0)$  na uogólnioną instrukcję odczytu  $\Downarrow(ra)$  mającą za zadanie wyznaczenie charakterystyki odniesienia  $S_R$  zgodnie z (1)”. Zastosowanie tej procedury do testu MATS+ prowadzi do następującej postaci transparentnej testu:  $\{\Downarrow(ra);\Uparrow(ra,w\bar{a});\Downarrow(r\bar{a},wa)\}$ . Za wyznaczenie sygnatury odniesienia odpowiedzialna jest w tym wypadku pierwsza faza testu. Należy również podkreślić, że faza wyznaczania sygnatury odniesienia ma tutaj charakter uniwersalny i jest identyczna dla każdego standardowego testu krokowego. Porównanie złożoności transparentnych wersji wybranych testów krokowych realizowanych techniką tradycyjną i techniką SAODC zawiera tab. 1.

Tab. 1. Porównanie złożoności techniki tradycyjnej i techniki SAODC

Test	Wersja bazowa	Technika tradycyjna		Technika SAODC	
		Sygnatura	Razem	Sygnatura	Razem
MATS	3N	2N	5N	1N	4N
MATS+	4N	2N	6N	1N	5N
MATS++	5N	3N	8N	1N	6N
March X	5N	3N	8N	1N	6N
March Y	7N	5N	12N	1N	8N
March A	14N	4N	18N	1N	15N
March B	16N	6N	22N	1N	17N
March C	10N	6N	16N	1N	11N
March C-	9N	5N	14N	1N	10N

Przedstawione wyniki dość wyraźnie pokazują zmniejszenie złożoności testów transparentnych realizowanych z użyciem techniki SAODC w stosunku do testów realizowanych techniką tradycyjną.

#### 4. BADANIA

Zaproponowane podejście do transparentnego testowania pamięci wykorzystujące charakterystykę adresową oprócz zmniejszenia złożoności testu pozwala również zwiększyć wydajność testu w odniesieniu do implementacji tradycyjnej. Można to osiągnąć poprzez porównanie wartości sygnatur nie na końcu testu, jak to ma miejsce w tradycyjnym podejściu testowania transparentnego, ale po każdej jego fazie. Jak zostało stwierdzone w punkcie poprzednim tradycyjne testy krokowe zmieniają zawartość pamięci z postaci  $D$  na  $D^*$  i odwrotnie. Zatem po zakończeniu każdej fazy aktualna zawartość pamięci (w wypadku pamięci działającej poprawnie) to  $D$  lub  $D^*$ . Na podstawie (2) wiadomo, iż w wypadku pamięci o rozmiarze  $N=2^m-1$  sygnatura odniesienia wyliczona na podstawie zawartości  $D$  jest równa sygnaturze wyznaczonej na podstawie zawartości  $D^*$ . Zatem nic nie stoi na przeszkodzie aby sygnatura robocza była wyznaczana w każdej fazie testu i porównywana z sygnaturą odniesienia  $S_R$  po każdej zakończonej fazie. Podejście takie pozwala zmniejszyć liczbę maskowań uszkodzeń jakie mogą mieć miejsce w podejściu tradycyjnym, a tym samym zwiększyć końcową wydajność procesu testowania. Niejasność może się pojawić w wypadku testów w których fazy zawierają więcej niż jedno polecenie odczytu, np. March B:  $\{\uparrow(w0); \uparrow(r0,w1,r1,w0,r0,w1); \uparrow(r1,w0,w1); \downarrow(r1,w0,w1,w0); \downarrow(r0,w1,w0)\}$ . Faza druga powyższego testu zawiera trzy polecenia odczytu. W wersji transparentnej będzie miała postać:  $\uparrow(ra,w\bar{a},r\bar{a},wa,ra,w\bar{a})$  lub po wykluczeniu wszystkich poleceń zapisu  $\uparrow(ra,r\bar{a},ra)$ . Może powstać pytanie, którego polecenia odczytu użyć do wyznaczenia sygnatury roboczej? Jest oczywiste, iż w wypadku uszkodzeń złożonych (np. CF) pierwsze polecenie odczytu ma większe znaczenie niż pozostałe i to ono powinno być traktowane w sposób priorytetowy. Odczytana wartość w pierwszym poleceniu może służyć jako wartość odniesienia (porównawcza) dla kolejnych poleceń występujących w danej fazie.

Poniżej przedstawione jest porównanie wydajności testów transparentnych realizowanych z zastosowaniem techniki SAODC w odniesieniu do testów realizowanych techniką standardową. Badania przeprowadzone zostały dla wielokrotnych uszkodzeń

sklejeniowych SAF do wykrywania których użyty został test MATS+. W realizacji standardowej jako wielomian wyliczania sygnatur przyjęto wielomian postaci:  $\varphi(x)=x^3+x+1$ .

Tab.2. Porównanie wydajności techniki tradycyjnej i techniki SAODC

Liczba uszkodzeń SAF	Wydajność testu	
	schemat tradycyjny [%]	schemat SAODC [%]
1	100	100
2	87,01	100
3	87,15	98,36
4	87,57	98,36
5	87,51	98,44

Z wyników przedstawionych w tab. 2 widać, iż nawet w wypadku tak prostego testu jakim jest MATS+ testowanie wykorzystujące charakterystykę adresową pamięci pozwala w sposób znaczący zwiększyć wydajność testu. W wypadku uszkodzeń podwójnych i przyjętego wielomianu, tradycyjne podejście pozwala wykryć około 87% sytuacji błędnie działającej pamięci, podczas gdy podejście oparte o SAODC i sprawdzanie sygnatury po każdej fazie testu pozwala wykryć 100% takich przypadków. Różnicę w efektywności można również zauważyć w wypadku wielokrotnych uszkodzeń SAF, gdzie liczba występujących jednocześnie uszkodzeń jest większa od 2. Również tutaj przedstawiana technika testowania okazuje się w sposób zauważalny bardziej efektywna od podejścia tradycyjnego.

## 5. PODSUMOWANIE

W artykule zaproponowane zostało wykorzystanie techniki testowania pamięci opartej o charakterystykę adresową (Self-Adjusting Output Data Compression – SAODC) [13] do testowania transparentnego. Zostały wskazane zalety powyższego podejścia w stosunku do podejścia tradycyjnego. Podejście wykorzystujące technikę SAODC umożliwia skrócenie testu, jak również w prosty sposób zwiększenie jego efektywności. Skrócenie testu dokonuje się poprzez ograniczenie do jednej instrukcji fazy wyliczania sygnatury odniesienia, zaś zwiększenie efektywności realizowane jest poprzez sprawdzenie wartości sygnatury po każdej fazie testu krokowego.

## 6. BIBLIOGRAFIA

- [1] Bardell P., McAnney W., Savir J.: Built In Test for VLSI: Pseudorandom Techniques, New York: John Wiley & Sons, 1987
- [2] Dekker, R., Beenker, F., and Thijssen, L.: Realistic Built-In Self-Test for Static RAMs. IEEE Des. Test Vol. 6, No. 1, 1989, pp. 26-34
- [3] Jain, S., Stroud, C.: Built-in Self Testing of Embedded Memories. IEEE Des. Test Vol. 3, No. 5, 1986, pp. 27-37
- [4] Rajsuman, R.: RAMBIST builder: a methodology for automatic built-in self-test design of embedded RAMs. In Proceedings of the IEEE international Workshop on Memory

- Technology, Design and Testing (MTDT '96) (August 13 - 14, 1996). MTDT. IEEE Computer Society, Washington, DC, p. 50
- [5] Treuer, R. and Agarwal, V. K.: Built-In Self-Diagnosis for Repairable Embedded RAMs. IEEE Des. Test Vol. 10, No. 2, 1993, pp. 24-33
- [6] Chen T., Sunada G.: Design of a self-testing and self-repairing structure for highly hierarchical ultra-large capacity memory chips. IEEE Trans. VLSI Syst. Vol. 1 No. 2: 1993, pp. 88-97
- [7] Le K. T., Saluja K. K.: A Novel Approach for Testing Memories Using a Built-In Self Testing Technique. In Proceedings of the IEEE International Test Conference, Washington, DC, 1986, pp. 830-839
- [8] Van de Goor A. J.: Testing Semiconductor Memories, Theory and Practice, Chichester, John Wiley & Sons, 1991
- [9] Nicolaidis M.: Transparent BIST for RAMs. In Proceedings of the IEEE International Test Conference, Baltimore, MD, 1992, pp. 598-607
- [10] Cockburn, B. F., Sat, Y. N.: Synthesized Transparent BIST for Detecting Scrambled Pattern-Sensitive Faults in RAMs. In Proceedings of the IEEE international Test Conference on Driving Down the Cost of Test Washington, DC, 1995, pp. 23-32.
- [11] Yarmolik V. N., Murashko I.A., Kummert A. Ivaniuk A.A.: Transparent testing of Digital Memories, Minsk Bielarus, Bestprint, 2005, 230p.
- [12] Hellebrand S., Wunderlich H.J., Yarmolik V. N.: Symmetric transparent BIST for RAMs. DATE'99: In Proceedings of the conference on Design, Automation and Test in Europe, pp. 135, New York, NY, USA, 1999. ACM Press
- [13] Yarmolik, V. N., Hellebrand, S., and Wunderlich, H.: Self-adjusting output data compression: an efficient BIST technique for RAMs. In Proceedings of the Conference on Design, Automation and Test in Europe Paris 1998. Design, Automation, and Test in Europe. IEEE Computer Society, Washington, DC, pp. 173-179
- [14] Sosnowski J.: Testowanie i niezawodność systemów komputerowych.: Akademicka oficyna wydawnicza EXIT, 2005.