

Bartosz SOKÓŁ¹, S.V. YARMOLIK²

¹POLITECHNIKA BIAŁOSTOCKA, WYDZIAŁ INFORMATYKI

²BELARUSIAN STATE UNIV. OF INFORMATICS AND RADIOEL

Optymalne adresowanie pamięci przy wykorzystaniu usterek uwarunkowanych zawartością (PSF)

Mgr inż. Bartosz SOKÓŁ

Ukończył Wydział Informatyki Politechniki Białostockiej w 2001 roku. Od tego czasu pracuje w Katedrze Systemów Komputerowych i zajmuje się metodami testowania pamięci oraz prowadzi zajęcia dydaktyczne z Testowania Układów Cyfrowych, Kryptografii i Ochrony Danych oraz Metod Optymalizacji.



e-mail: bsokol@ii.pb.bialystok.pl

S.V. YARMOLIK

Studentka piątego roku Uniwersytetu Białoruskiego Informatyki i Radioelektroniki. Zajmuje się badaniami nad metodami i technikami transparentnego testowania pamięci.



e-mail: yarmolik@bsuir.unibel.by

Streszczenie

Celem niniejszej publikacji jest zaprezentowanie nowych algorytmów i technik testowania pamięci bazując na transparentnych testach pamięci. Niniejsza publikacja dotyczy problemu wykrywania usterek pamięci uwarunkowanych zawartością (ang. pattern sensitive faults - PSF). Zawarte w pracy informacje pokazują efektywność użycia testów krokowych przy wykrywaniu pasywnych usterek PSF (passive pattern sensitive faults PPSF). Zbadano efektywność wielokrotnego użycia testów krokowych do wykrywania wyżej wymienionych usterek PPSF oraz przeanalizowano optymalny wybór adresu startowego.

Słowa kluczowe: usterki uwarunkowane zawartością, transparentne testy krokowe, adresy początkowe.

Optimal Memory Address Seeds for Pattern Sensitive Faults Detection

Abstract

The goal of this paper is to propose new algorithms and techniques for memory testing based on transparent memory tests. This paper deals with memory pattern sensitive faults detection problem. It shows the efficiency of using march tests for detection memory passive pattern sensitive faults. Also it shows the efficiency of multiple runs of march tests for detection PPSF's and analyzes the optimal address seeds.

Keywords: pattern sensitive faults, transparent memory tests, address seeds.

1. Wstęp

Wszystkie usterki pamięci mogą być klasyfikowane w zależności od liczby komórek zaangażowanych w daną usterkę, na usterki jednokomórkowe (np. usterki sklejniowe (*stuck-at faults*), usterki przejścia (*transition faults*)) oraz usterki wielokomórkowe (np. usterki sprzężenia (*coupling faults*)) które są znacznie trudniejsze do wykrycia. Ogólnym przypadkiem usterek należących do drugiej grupy są usterki uwarunkowane zawartością (*Pattern Sensitive Faults - PSF*). Komórka zwana komórką bazową (*base cell*) zawiera usterkę PSF, jeżeli wartość w tej komórce zmienia się w wyniku otrzymania wartości „0” lub „1” lub przejść „0→1” lub „1→0” w grupie pozostałych komórek nazywanych komórkami wiążącymi (*neighbourhood cell's*). Istnieją trzy typy usterek PSF: **Pasywne PSF (PPSF)**: zawartość komórki bazowej nie może zostać zmieniona z powodu ustalonych wartości w komórkach wiążących. **Statyczne PSF (SPSF)**: zawartość komórki bazowej jest wymuszana przez ustaloną zawartość komórek wiążących. **Aktywne PSF (APSF)**: komórka bazowa zmienia swoją zawartość w zależności od zmian w komórkach wiążących. Zmiana taka może składać się z przejścia (*transition*) w jednej komórce wiążącej, podczas gdy pozostałe komórki wiążące i komórka bazowa zawierają ustaloną wartość. Usterki PSF powstają głównie z powodu dużej gęstości i upakowania układów oraz wzajemnych zależności pomiędzy sygnałami.

Ponieważ wraz ze wzrostem gęstości pamięci RAM, komórki umieszczane są coraz bliżej siebie, co powoduje, że usterki PSF zaczynają dominować pośród wszystkich występujących usterek. Ponadto, pozostałe typy usterek takie jak *stuck-at faults*, *transition faults* a także *coupling faults* mogą być uważane jako specjalny typ usterek PSF.

2. Analiza zdolności testów krokowych do wykrywania usterek pamięci

Podczas naszych badań skoncentrujemy się na usterkach *Passive PSF* – jako najtrudniejszych do wykrycia. Każda komórka pamięci, spośród N komórek może być zaangażowana w usterkę PPSF. Notacja ta oznacza, że k przypadkowych komórek pamięci jest zaangażowanych w usterkę PPSF a jedną z tych komórek jest **komórka bazowa (base cell - b)**, pozostałe $k-1$ komórek to **komórki wiążące (neighbors - n)**. Istnieje k klas różnych usterek PPSF. Klasyfikacja ta zależy od uporządkowania w przestrzeni adresów oraz pozycji wszystkich tych komórek w tej przestrzeni. Niech adresy pamięci $i_0, i_1, i_2, \dots, i_{k-1}$, dla konkretnej usterki PPSF będą posortowane w porządku wzrastającym w sposób następujący $i_0 < i_1 < i_2 < \dots < i_{k-1}$. Wtedy każda usterka PPSF będzie reprezentowana przez elementy $a_{i_0}, a_{i_1}, a_{i_2}, \dots, a_{i_{k-1}}$ uporządkowane w przestrzeni adresów zgodnie ze wzrastającym porządkiem adresów komórek pamięci. Jedną spośród k komórek jest komórką bazową, więc istnieje k różnych usterek PPSF zależnych od pozycji komórki bazowej. Oznacza to, że istnieje k klas usterek PPSF zależnych od pozycji komórki bazowej. Na przykład dla $k = 5$ mamy pięć następujących klas: $b_{i_0}, n_{i_1}, n_{i_2}, n_{i_3}, n_{i_4}$; $n_{i_0}, b_{i_1}, n_{i_2}, n_{i_3}, n_{i_4}$; $n_{i_0}, n_{i_1}, b_{i_2}, n_{i_3}, n_{i_4}$; $n_{i_0}, n_{i_1}, n_{i_2}, b_{i_3}, n_{i_4}$; $n_{i_0}, n_{i_1}, n_{i_2}, n_{i_3}, b_{i_4}$. Istnieje 2^{k-1} różnych kombinacji bitów dla zbioru bitów w komórkach wiążących. Dokładna liczba usterek PPSF określona jest zgodnie z równaniem:

$$L(PPSFk) = k2^{k-1} \times \binom{N}{k} \quad (1)$$

Ten sam rezultat otrzymujemy na podstawie następującej obserwacji. Komórka bazowa może przyjąć każdą spośród N możliwych pozycji wśród komórek pamięci, a dla dowolnej kombinacji bitów w komórkach wiążących istnieje 2^{k-1} różnych zbiorów bitów. Mamy, więc:

$$L(PPSFk) = N \times 2^{k-1} \times \binom{N-1}{k-1} = k2^{k-1} \times \binom{N}{k} \quad (2)$$

Ostatnie równanie pozwala na wyciągnięcie wniosku, iż wśród wszystkich k klas danej usterki istnieje taka sama liczba usterek PPSF.

Sprawdźmy kilka testów pamięci pod kątem ich zdolności do wykrywania usterek PPSF. Większa część testów pamięci posiada sekwencje działań umieszczone w fazach. Wszystkie fazy

składają się z operacji odczytu (*read* – *r*) i zapisu (*write* – *w*). Na przykład test MATS+ posiada następującą postać:

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

a jego wersja transparentna wygląda następująco:

$$\{\uparrow\downarrow(ra, wa^*); \downarrow(ra^*, wa)\},$$

gdzie $a \in \{0, 1\}$. Przypuśćmy, że używamy tego testu do testowania 8-mio bitowej pamięci, w której zawartość początkowa (*background*) jest równa samym zerom $A = a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7 = 00000000$. Wtedy kolejne stany testowanej pamięci przy wykorzystaniu testu MATS+ będą wyglądały tak jak te przedstawione w tabeli 1.

Tab. 1. Zawartość pamięci podczas testowania testem MATS+

Fazy testu MATS+	Zawartość pamięci							
	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
$\uparrow\downarrow(ra, wa^*)$	0	0	0	0	0	0	0	0
	1	1	0	0	0	0	0	0
	1	1	1	0	0	0	0	0
	1	1	1	1	0	0	0	0
	1	1	1	1	1	0	0	0
	1	1	1	1	1	1	0	0
	1	1	1	1	1	1	1	0
	1	1	1	1	1	1	1	1
$\downarrow\downarrow(ra^*, wa)$	1	1	1	1	1	1	1	0
	1	1	1	1	1	1	0	0
	1	1	1	1	1	0	0	0
	1	1	1	1	0	0	0	0
	1	1	1	0	0	0	0	0
	1	1	0	0	0	0	0	0
	1	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0

Należy zauważyć, że sekwencja adresów pamięci została wybrana w oparciu o sekwencję licznikową, a adresem startowym był adres $i_0 = 0$. Jak widać w powyższej tabeli, tylko jeden zbiór bitów, spośród wszystkich możliwych dla każdej aktywnej komórki, został zaznaczony pogrubioną i pochyloną czcionką. W rzeczywistości sprawdzamy (odczyt 0 zapis 1 podczas pierwszej fazy i odczyt 1 zapis 0 podczas drugiej fazy) komórkę a_0 w obu fazach dla tej samej zawartości w pozostałych komórkach. Aktywacja usterki *PPSFk* może wystąpić tylko w momencie sprawdzania komórki bazowej. Podsumowując widać, że wykrycie usterki *PPSFk* jest możliwe tylko dla jednej kombinacji bitów w pozostałych $k-1$ komórkach wiążących spośród 2^{k-1} możliwych kombinacji. Dlatego też liczba $Q1(PPSFk(i_0))$ usterek możliwych do wykrycia podczas pierwszego uruchomienia testu MATS+ wynosi:

$$Q1(PPSFk(i_0)) = k \times \binom{N}{k} \quad (3)$$

A pokrycie usterek (*fault coverage FC*) dla testu MATS+ wynosi:

$$FC1_MATS+(PPSFk(i_0)) = \frac{Q1(PPSFk(i_0))}{L(PPSFk)} 100\% = \frac{1}{2^{k-1}} 100\%. \quad (4)$$

Na przykład: $FC_MATS+(PPFS5) = (1/2^4) 100\% = 6.25\%$. Pokrycie usterek opisane wzorem (4) jest poprawne dla każdego testu pamięci, z kolejnymi fazami ustawionymi tak jak w przypadku testu MATS+, mianowicie $\{\dots \uparrow\downarrow(ra, \dots, wa^*); \downarrow\downarrow(ra^*, \dots, wa); \dots\}$. W celu podniesienia możliwości wykrycia usterek *PSF* musimy zastosować test z czterema kolejnymi fazami $\{\dots \uparrow\downarrow(ra^*, \dots, wa); \uparrow\downarrow\downarrow(ra, \dots); \dots \downarrow\downarrow(ra^*, \dots, wa); \uparrow\downarrow\downarrow(ra, \dots); \dots\}$ (na przykład test March C). W praktyce wystarczy mniej niż cztery fazy. Zaprezentowane pary faz są potrzebne do aktywowania usterki a w rezultacie do jej wykrycia.

Szybka analiza możliwych do osiągnięcia wyników pozwala na wyciągnięcie wniosku, że dla każdej komórki bazowej, usterki *PPSFk* będą aktywowane przez dwie różne zawartości pamięci w ramach całej pamięci z $N-1$ komórkami. Więc dla testu zawierającego kolejne fazy: $\{\dots \uparrow\downarrow(ra, \dots, wa^*); \uparrow\downarrow\downarrow(ra^*, \dots, wa); \downarrow\downarrow(ra, \dots); \dots\}$,

niech będzie to test March C, liczba $Q2(PPSFk(i_0))$ wykrywanych usterek podczas pierwszego uruchomienia testu wynosi:

$$Q2(PPSFk(i_0)) = 2k \times \binom{N}{k} \quad (5)$$

A pokrycie usterek (*FC*) dla testu March C wynosi:

$$FC2_March_C(PPSFk(i_0)) = \frac{Q2(PPSFk(i_0))}{L(PPSFk)} 100\% = \frac{1}{2^{k-2}} 100\%. \quad (6)$$

W przypadku usterki *PPFS5* i testu March C mamy: $FC_March_C(PPFS5) = (1/2^3) 100\% = 12.5\%$.

Oczywiście, dla takiej samej zawartości pamięci, każdy test pamięci będzie wykrywał takie same usterki pamięci a także taką samą ich ilość.

Bazując na dwóch powyższych przykładach, możemy rozważyć dwa różne przypadki. Przypadek pierwszy, gdy komórka bazowa posiada stałą pozycję a komórki wiążące są na przypadkowych pozycjach w ramach $N-1$ komórek pamięci RAM. Drugi przypadek jest bardziej realistyczny, gdzie zarówno komórka bazowa, jak też komórki wiążące znajdują się na losowych pozycjach w ramach N -bitowej pamięci. Dla powyższych przypadków możemy sformułować kolejny problem. Jak zwiększyć pokrycie uszkodzeń bazując na krokowych testach pamięci i stałej zawartości?

3. Dobór adresów początkowych przy wykrywaniu usterek pattern sensitive faults

Odpowiadając na postawione powyżej pytanie, wiadomo, że dla stałych dwóch elementów: wybranego testu krokowego oraz zawartości pamięci podczas procedury testowania, jedyną możliwością zwiększenia pokrycia usterek jest wykorzystanie procedury generowania adresów. Zaczniemy od optymalizacji wartości adresu startowego. Rozważmy przypadek procedury testowej z wielokrotnym uruchomieniem testu krokowego. Bazując na stwierdzeniu, że liczba usterek wykrywanych przez test krokowy nie zależy od porządku adresów, przy kolejnym uruchomieniu testu możemy użyć nowego adresu startowego. W tym samym czasie konfiguracja usterek może być częściowo różna. Jako przykład rozważmy drugie uruchomienie testu MATS+ z adresem początkowym równym $i_1 = 1$. Wtedy kolejne stany we wszystkich komórkach pamięci dla przykładu z tabeli 1 zostały przedstawione w tabeli 2.

Tab. 2. Zawartość pamięci podczas testowania testem MATS+ po zmianie adresu startowego

Fazy testu MATS+	Zawartość pamięci							
	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
$\uparrow\downarrow(ra, wa^*)$	0	1	0	0	0	0	0	0
	0	1	1	0	0	0	0	0
	0	1	1	1	0	0	0	0
	0	1	1	1	1	0	0	0
	0	1	1	1	1	1	0	0
	0	1	1	1	1	1	1	0
	0	1	1	1	1	1	1	1
	1	1	1	1	1	1	1	1
$\downarrow\downarrow(ra^*, wa)$	0	1	1	1	1	1	1	1
	0	1	1	1	1	1	1	0
	0	1	1	1	1	1	0	0
	0	1	1	1	1	0	0	0
	0	1	1	1	0	0	0	0
	0	1	0	0	0	0	0	0
	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0

Ponownie jak poprzednio otrzymaliśmy tylko jedną kombinację bitów w ramach całej pamięci dla każdej aktywnej komórki bazowej b . Porównując powyższe wyniki z przypadkiem opisanym w tabeli 1, widzimy, że zostały wygenerowane nowe kombinacje bitów. Dla każdej komórki bazowej b , z wyjątkiem przypadku, gdy $b = a_0$, istnieją różne wartości w komórce a_0 . Dla przypadku, gdy $b = a_0$ mamy $N-1$ różnych wartości w zbiorze bitów w komórkach wiążących. Widzimy, że liczba wykrywanych usterek

PPSFk jest taka sama w obu przypadkach, istnieje jednak grupa usterek wykrywanych podczas pierwszego uruchomienia z adresem i_0 oraz nowe usterek *PPSFk* wykrywane tylko podczas drugiego uruchomienia.

Na przykład, gdy komórka bazowa $b = a_i$ a adres początkowy $i_0 = 0$ (patrz tabela 1) mamy dwie klasy usterek *PPSF5*. Jest to 15 usterek $b_{i_0, n_{i_1}, n_{i_2}, n_{i_3}, n_{i_4}} = b0000$ oraz 20 usterek $n_{i_0, b_{i_1}, n_{i_2}, n_{i_3}, n_{i_4}} = 1b000$. Dla nowego adresu początkowego $i_1 = 1$ (patrz tabela 2) mamy 15 usterek $b_{i_0, n_{i_1}, n_{i_2}, n_{i_3}, n_{i_4}} = b0000$ oraz 20 nowych usterek $n_{i_0, b_{i_1}, n_{i_2}, n_{i_3}, n_{i_4}} = 0b000$ nie wykrytych dotychczas podczas pierwszego uruchomienia testu MATS+.

Jeżeli komórka bazowa będzie miała inną pozycję, na przykład $b = a_3$, wtedy dla adresu początkowego $i_0 = 0$ mamy cztery klasy usterek *PPSF5*, mianowicie jedną usterek $b_{i_0, n_{i_1}, n_{i_2}, n_{i_3}, n_{i_4}} = b0000$, 12 usterek $n_{i_0, b_{i_1}, n_{i_2}, n_{i_3}, n_{i_4}} = 1b000$, 18 usterek $n_{i_0, n_{i_1}, b_{i_2}, n_{i_3}, n_{i_4}} = 11b00$ oraz 4 usterek $n_{i_0, n_{i_1}, n_{i_2}, b_{i_3}, n_{i_4}} = 111b0$. Z nowym adresem początkowym $i_1 = 1$ podział usterek będzie inny (patrz tabela 2).

Całkowicie inna sytuacja występuje w przypadku, gdy komórka bazowa jest równa $b = a_0$ a adres początkowy wynosi $i_1 = 1$. W tym przypadku wszystkie usterek wykrywane podczas drugiego uruchomienia testu będą inne w porównaniu z usterkami wykrytymi podczas pierwszego uruchomienia testu MATS+ dla przypadku $i_0 = 0$. Jak widać, podczas pierwszego uruchomienia wykrywane są usterek $b0000$ zaś podczas drugiego uruchomienia wykrywane są tylko usterek $b1111$.

Biorąc pod uwagę wszystkie możliwe pozycje komórki bazowej, całkowita liczba nowych usterek *PPSF5* z nową adresacją i_1 wynosi 175.

Jak łatwo pokazać, dla ogólnego przypadku, liczba dodatkowych usterek *PPSFk* wykrywanych podczas drugiego uruchomienia testu MATS+ z adresacją $i_1 = 1$ może być oszacowana jako:

$$(N-1) \binom{N-2}{k-2} + \binom{N-1}{k-1} \quad (7)$$

Dla $i_1=2$ łatwo wykazać, że:

$$(N-2) \times \left[\binom{2}{1} \times \binom{N-3}{k-2} + \binom{2}{2} \times \binom{N-3}{k-3} \right] + 2 \times \left[\binom{N-2}{k-1} + \binom{N-2}{k-2} \right] \quad (8)$$

Należy zaznaczyć, iż nowa adresacja pamięci i_1 dzieli wszystkie możliwe (N) komórki bazowe na dwie grupy. Podział jest wykonywany zgodnie z odległością $s = i_1 - i_0$ pomiędzy dwoma adresami startowymi (należy pamiętać, że $i_1 > i_0$). Zgodnie z tymi obliczeniami, całkowity zbiór wszystkich możliwych stałych pozycji komórki bazowej będzie podzielony na dwie grupy. Pierwsza grupa będzie miała s różnych bitów pośród $N-1$ komórek pamięci (z wyjątkiem pozycji komórki bazowej) wygenerowanych podczas drugiego uruchomienia testu MATS+ w porównaniu z kombinacjami bitów otrzymanymi podczas pierwszego uruchomienia z adresem początkowym $i_0 = 0$. Liczba takich komórek bazowych jest określana przez $N-s$. Reszta komórek bazowych należy do drugiego zbioru z $N-s$ różnymi bitami.

Jako przykład weźmy $i_1 = 3$. Dla tego przypadku mamy $s = i_1 - i_0 = 3 - 0 = 3$. Więc istnieje $N-s = 8 - 3 = 5$ komórek bazowych z $s = 3$ różnymi bitami w ramach wszystkich $N-1 = 8 - 1 = 7$ komórek pamięci, jak to zostało pokazane w tabeli 2.

4. Podsumowanie

Nowe zbiory bitów będą generowane w komórkach wiążących dla każdej możliwej komórki bazowej, dzięki czemu powinny być wykrywane nowe usterek *PPSFk*. Liczba nowych usterek *PPSFk* może być określona przy pomocy odległości Hamminga pomiędzy zawartościami pamięci dla dwóch różnych adresów początkowych. W naszym przypadku odległość $s = i_1 - i_0$ pomiędzy dwoma adresami startowymi oraz wartość $N-s = (i_1 - i_0)$ równa się odległości Hamminga dla zbiorów bitów w komórkach wiążących dla tej samej komórki bazowej przy dwóch uruchomieniach testu MATS+. Na przykład, jeżeli odległość $s = i_1 - i_0 = 2$, oznacza to,

że dla $i_0 = 0$, $i_1 = 2$, oraz dla każdego i_0 i i_1 gdzie $i_0 - i_1 = 2$ mamy 6 ($N-s = 8-2 = 6$) par zbiorów bitów: $((11b00000, 00b000000), (111b0000, 001b0000), (1111b000, 0011b000), (11111b00, 00111b00), (111111b0, 001111b0), (1111111b, 0011111b))$, dla których odległość Hamminga wynosi $s = 2$, oraz mamy $s = 2$ pary zbiorów bitów $((b0000000, b0111111), (1b000000, 1b011111))$ z odległością $N-2 = 6$.

Dokładna liczba nowych usterek *PPSFk* wykrytych podczas drugiego uruchomienia testu MATS+ dla $s > 1$ będzie wyliczona zgodnie z równaniem:

$$Q1(PPSFk(i_1)) = (N-s) \times \sum_{i=1}^{\min(s, (k-1))} \binom{s}{i} \binom{N-s-1}{k-i-1} + s \times \sum_{i=1}^{\min(s, (k-1))} \binom{s-1}{i-1} \binom{N-s}{k-i} \quad (9)$$

Analiza powyższego równania pozwala na wysunięcie następujących wniosków. Aby osiągnąć wysokie pokrycie usterek podczas drugiego uruchomienia testu MATS+ musimy użyć porządku adresów i_1 :

1. Pierwszy przypadek (stała pozycja komórki bazowej) – musimy otrzymać maksymalną odległość Hamminga pomiędzy zbiorami bitów w komórkach wiążących dla stałej pozycji komórki bazowej. Aby osiągnąć ten cel, nowy adres początkowy, przy drugim uruchomieniu testu pamięci, powinien spełniać następujące równanie $i_1 - i_0 = b + 1$, gdzie b jest to numer komórki bazowej.
2. W drugim przypadku (komórka bazowa może przyjąć dowolną pozycję losową) musimy otrzymać, o ile to możliwe, maksymalną sumę S_{HD} odległości Hamminga dla wszystkich pozycji komórki bazowej. Liczba ta może być policzona jako

$$S_{HD} = s(N-s) + (N-s)s \quad (10)$$

Maksimum dla funkcji S_{HD} może być wyliczone z równania:

$$\frac{\partial(S_{HD})}{\partial s} = \frac{\partial(s(N-s) + (N-s)s)}{\partial s} = (N - \frac{s}{2}) = 0, \quad (11)$$

gdzie wynikiem jest $s = N/2$.

Wracając do naszego przykładu, dla wersji pierwszej mamy $i_1 - i_0 = s = b$. Oczywiście jest to najlepsze rozwiązanie pozwalające na zwiększenie pokrycia usterek. Na przykład, jeżeli ustaloną pozycją komórki bazowej jest 3, wtedy dla $i_0 = 0$ mamy $111b0000$. Wtedy z równania $i_1 - i_0 = b + 1 = 3 + 1 = 4$ możemy wziąć $i_1 = 4$.

Łatwo można wykazać, iż zaproponowane techniki mogą być rozszerzone na wielokrotne użycie testu pamięci z wykorzystaniem różnych adresów początkowych: $i_0, i_1, i_2, i_3, \dots$

5. Literatura

- [1] A.J. van de Goor. Testing Semiconductor Memories, Theory and Practice. - Chichester UK, John Wiley & Sons, 1991.
- [2] L. Cheng., C.W. Wu "Neighborhood Pattern-Sensitive Fault Testing for Semiconductor Memories", proc. VLSI Design/CAD Symp., Pingtung Aug. 2000, s. 401-404.
- [3] M.G.Karpovsky, V.N. Yarmolik, "Transparent Memory Testing for Pattern Sensitive Faults", IEEE International Test Conference, 1994., s. 860-869.
- [4] M.Nicolaidis, "Theory of Transparent BIST for RAMs", IEEE Transaction of Computers, vol. 45, N. 10, 1996.
- [5] V.N.Yarmolik, B.Sokol, S.V.Yarmolik, "Counter sequences for memory test address generation", Mixed Design of Integrated Circuits and Systems, proc. 12th International Conference MIXDES 2005, Krakow, Poland 22-25 June, 2005, s.413-418.
- [6] Sokol B., Mrozek I., Yarmolik V.N., "Transparent March Tests to effective Pattern Sensitive Faults Detection", EAST-WEST DESIGN & TEST WORKSHOP - EWDWTW'04, Jalta, Ukraina 22-26 Sept. 2004, s. 166-171.