

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

## GENEROWANIE OPTIMALNYCH WEKTORÓW TESTOWYCH PAMIĘCI RAM<sup>2</sup>

*Efektywne wektory testowe i sposoby ich generowania odgrywają znaczącą rolę w testowaniu zarówno oprogramowania jak i sprzętu. Jednym z najtańszych podejść do problemu generowania wektorów testowych jest podejście oparte o wektory losowe. Obecnie w literaturze zaproponowanych zostało wiele nowych metod znacząco poprawiających efektywność testowania losowego należących do grupy kontrolowanego testowania losowego. W podejściu tym wybrane parametry tworzonych wektorów są z góry ustalane. W artykule zaproponowana zostanie nowa metoda generowania wektorów testowych. W proponowanym podejściu jako funkcja ich dopasowania użyta zostanie Maksymalizacja Minimalnej Odległości Hamminga, w odróżnieniu do standardowej odległości Hamminga używanej w podejściu znanym z literatury. Pozwoli to na wygenerowanie optymalnych wektorów testowych, których efektywność sprawdzona zostanie w odniesieniu do uszkodzeń PSF pamięci RAM.*

## OPTIMAL TEST VECTORS FOR RAM TESTING

*Efficient test vectors and methods of their generation play crucial role in both hardware and software testing. Random testing is a low cost method that can be applied to a wide range of testing problems. Recently, a number of novel techniques, include Antirandom Testing strategies, have been developed to increase the effectiveness of random testing. In this paper, we propose a new algorithm for antirandom test generation. As the fitness function we use Maximal Minimal Hamming Distance (MMHD) rather than standard Hamming distance as is used in the classical approach. This approach allows us to generate optimal test vectors for RAM testing.*

### 1. WSTĘP

Jednym z najbardziej widocznych dzisiaj przejawów gigantycznego postępu naukowo-technicznego jest stopień rozwoju elektroniki cyfrowej. Jednym z efektów tego rozwoju, a może bardziej jego przyczyną, jest wciąż rosnące zapotrzebowanie na szeroko rozumiane systemy cyfrowe. Nierozzerwalnie z elektroniką cyfrową związane są pamięci

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

<sup>2</sup> Badania finansowane z pracy statutowej S/WI/5/08 Wydziału Informatyki Politechniki Białostockiej

półprzewodnikowe DRAM, SRAM, ROM, EPROM itd. Praktyka pokazuje, iż pamięć zajmuje ponad 50% całej powierzchni krzemowej układu SoC (ang. System on a Chip) [1]. Statystycznie około 70% wszystkich uszkodzeń w systemach cyfrowych spowodowane jest uszkodzeniami pamięci

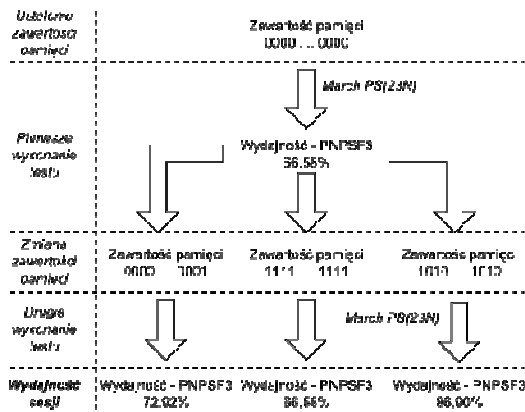
Kategorią uszkodzeń, które są jednymi z najbardziej trudnych do wykrycia w pamięci RAM, jest kategoria uszkodzeń uwarunkowanych zawartością (ang. PSF – Pattern Sensitive Faults). Uszkodzenia te wiążą zależnością grupę komórek pamięci. Jednoczesne wystąpienie odpowiednich wartości we wszystkich komórkach takiej grupy, może prowadzić do uaktywnienia tego uszkodzenia. Uaktywnienie przejawiać się będzie np. niemożnością dokonania zmiany wartości w jednej z komórek czy wymuszeniem w niej określonej wartości.

Dotychczasowe badania wykazały, iż przy obecnych rozmiarach pamięci jedynymi testami możliwymi do realnego zastosowania w procesie testowania pamięci są testy z grupy testów maszerujących (ang. march tests). Wykazują się one bardzo wysoką wydajnością szczególnie w odniesieniu do uszkodzeń prostych (np. uszkodzenia sklejenkowe). Problem pojawia się jednak w odniesieniu do uszkodzeń złożonych. Wtedy ich jakość (rozumiana jako liczba wykrywanych uszkodzeń danego typu w stosunku do wszystkich uszkodzeń tego typu) pozostawia wiele do życzenia. Dlatego zaproponowane zostały techniki umożliwiające zwiększenie efektywności wykrywania uszkodzeń PSF. Najważniejsze z nich to algorytm pokryciowy (ang. tiling method) [2], algorytm dwóch grup (ang. two-group method) [2], algorytm wierszowy (ang. row-March Algorithm) [4], testy transparentne (ang. transparent testing) [3,6], testowanie pseudowyczerpujące (ang. pseudo-exhaustive testing) [5], testowanie wieloprzebiegowe oparte o zmianę sekwencji adresowych [6] i testowanie wieloprzebiegowe oparte o zamianę początkowej zawartości pamięci [7]. Najbardziej obiecującą techniką z wymienionych powyżej wydaje się technika testowania wieloprzebiegowego oparta o zmianę zawartości początkowej pamięci (wektorów początkowych pamięci). Technika ta polega na wielokrotnej realizacji zadanego testu przy jednoczesnej zmianie, przed każdą kolejną iteracją testu, zawartości pamięci. Jednak w zależności od przyjętej zawartości pamięci wynik końcowy może ulegać dużym wahaniom. Przykład dla przypadku dwuprzebiegowej sesji testowej testu March PS(23N) i uszkodzenia PNPSF3 przedstawia rys. 1.

W artykule uwaga skupiona jest na doborze optymalnych wektorów pamięci (wektorów testowych), które będą gwarantowały najwyższe możliwe pokrycie uszkodzeń złożonych. Zostanie zaproponowany algorytm umożliwiający dobór optymalnych wektorów pamięci dla dwu, trzy i czteroprzebiegowych sesji testowych opartych o testy proste klasy MATS+.

## 2. TESTY KROKOWE

Kluczową rolę w technikach testowania pamięci odgrywają testy maszerujące, nazywane również testami krokowymi czy testami typu March [2]. Test typu March składa się ze skończonej liczby sekwencji elementów typu March. Element typu March składa się ze skończonej liczby sekwencji 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 i 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



Rys.1. Wpływ zmiany zawartości pamięci na jakość dwuprzbiegowej sesji testowej.

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  $\{ \dots \}$ , podczas gdy element typu March ograniczony jest parą nawiasów okrągłych  $( \dots )$ . Jako przykład rozpatrzmy test MATS+ [2]:

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

M0      M1      M2

Test ten składa się z trzech elementów typu March: M0, M1, M2. Element M0 zeruje pamięć. Element M1, w porządku adresów rosnących, odpowiednio czyta wartość 0 z komórki, następnie zapisuje tam wartość 1. Element M2 testu MATS+, w porządku adresów malejących odczytuje wartość 1 z komórki, po czym do tej samej komórki wpisuje wartość 0.

### 3. WPŁYW ODLEGŁOŚCI HAMMINGA NA WYDAJNOŚĆ SESJI TESTOWYCH

W [8] wykazane zostało, iż w wypadku testowania wieloprzbiegowego opartego o zmianę zawartości pamięci w poszczególnych iteracjach sesji testowej, duży wpływ na wydajność całej sesji ma odległość Hamminga pomiędzy poszczególnymi zawartościami początkowymi pamięci użytymi w kolejnych iteracjach. Sformułowane zostało twierdzenie uwzględniające powyższy fakt.

**Twierdzenie 1.** W przypadku sesji testowej opartej o  $m$  iteracji testu krokowego pozwalającego wygenerować tylko jeden wzorzec w danej grupie komórek, optymalny zbiór początkowych wektorów pamięci  $B_0, B_1, \dots, B_{m-1}$ , użytych w kolejnych iteracjach testu, charakteryzuje się maksymalną odległością Hamminga  $HD(B_k, B_j)$  pomiędzy dowolną parą  $(B_k, B_j)$  gdzie  $k, j \in \{0, 1, 2, \dots, m-1\}$ .

Wpływ ten został sprawdzony drogą symulacji w odniesieniu do trójprzebiegowej sesji testowej testu March PS(3N) i uszkodzenia PNPSF3. Otrzymane wyniki zamieszczone są w tab. 1.

Tab1. Wpływ odległości Hamminga na wydajność sesji testowej

Wektory pamięci	Odległość Hamminga	Wydajność
$B_0=00000000$ $B_1=00000001$ $B_2=00000010$	$HD(B_0, B_1)=1$ $HD(B_0, B_2)=1$ $HD(B_1, B_2)=2$	21,87 %
$B_0=00000000$ $B_1=00000111$ $B_2=10110000$	$HD(B_0, B_1)=3$ $HD(B_0, B_2)=3$ $HD(B_1, B_2)=6$	33,03 %
$B_0=00000000$ $B_1=11001111$ $B_2=11110010$	$HD(B_0, B_1)=6$ $HD(B_0, B_2)=5$ $HD(B_1, B_2)=5$	37,05 %

Z przedstawionych rezultatów wynika silny związek pomiędzy jakością sesji testowej a odległością Hamminga pomiędzy wektorami początkowymi pamięci użytymi w kolejnych iteracjach testu. W przedstawionych rezultatach wydajność sesji testowej zmienia się w zakresie od 21, 87% do 37, 05%. Można zauważyć, iż minimalna wydajność rozpatrywanej sesji testowej zachodzi dla przypadku, gdy odległość pomiędzy wszystkimi parami początkowych wektorów pamięci jest najmniejsza (odpowiednio  $HD(B_0, B_1) = 1$ ,  $HD(B_0, B_2) = 1$  i  $HD(B_1, B_2) = 2$ ). Maksymalna wartość wydajności rozpatrywanej sesji testowej (37, 05%) jest w przypadku, gdy odległość Hamminga pomiędzy wszystkimi parami początkowych wektorów pamięci jest największa (odpowiednio  $HD(B_0, B_1) = 6$ ,  $HD(B_0, B_2) = 5$  i  $HD(B_1, B_2) = 5$ ). Problem pojawia się jednak z wygenerowaniem wektorów pamięci charakteryzujących się maksymalną odległością Hamminga. Kwestia ta jest rozpatrywana dokładnie w kolejnym rozdziale dla przypadku dwu-, trzy- i czteroprzebiegowych sesji testowych.

#### 4. WIELOPRZEBIEGOWE SESJE TESTOWE

W wypadku dwuprzebiegowych sesji testowych Twierdzenie 1 może zostać zapisane jako:  $\max\{HD(B_i, B_j)\}$  dla  $\forall i \neq j \in \{1, 2, \dots, 2^N\}$ . Aby powyższy warunek był spełniony dla dwóch wektorów pamięci  $B_i$  i  $B_j$ , odległość pomiędzy tymi wektorami musi być równa  $HD(B_i, B_j) = N$ . Tak więc generacja drugiego wektora polega na wyznaczeniu wartości komplementarnej pierwszego wektora:  $B_j = B_i^* = I - B_i$ , gdzie  $I$  jest jednostkowym wektorem  $N$ -wymiarowym, w którym wszystkie elementy równe są 1. Można jednak zauważyć, iż maksymalna wydajność dwuprzebiegowej sesji testowej jest również osiągana na bazie wektorów  $B_i, B_j$  w których  $B_j \triangleleft B_i^*$ . Okazuje się, iż maksymalną wydajność dwuprzebiegowej sesji testowej dla uszkodzeń PNPSFk [2] można osiągnąć dla wektorów  $B_i = b_{i1}b_{i2} \dots b_{iN}$ ,  $B_j = b_{j1}b_{j2} \dots b_{jN}$  spełniających nierówność (1).

$$HD(B_i, B_j) > N - k \quad (1)$$

Biorąc pod uwagę specyfikę działania testów generujących tylko pojedynczy wzorzec w pamięci (np. MATS+), można łatwo pokazać, iż przy teście dwuprzebiegowym i spełnieniu warunku opisanego nierównością (1), wektory wygenerowane dla dowolnych  $k$  komórek w obu przebiegach testu nie ulegną powtórzeniu.

Biorąc pod uwagę Twierdzenie 1, w wypadku trójprzebiegowych sesji testowych, należy wyznaczyć maksymalną z minimalnych wartości odległości Hamminga pomiędzy dowolną parą  $(B_i, B_j)$ ,  $(B_i, B_l)$ ,  $(B_j, B_l)$  trzech wektorów pamięci  $\{B_i, B_j, B_l\} \forall i \neq j \neq l \in \{1, 2, \dots, 2^N\}$ . Problem ten może zostać sformułowany jako problem wyznaczenia maksymalnej z minimalnych odległości Hamminga, który matematycznie można zdefiniować wzorem (2).

$$MMHD(B_i, B_j, B_l) = \underset{\forall i \neq j \neq l \in \{1, 2, 3, \dots, 2^N\}}{MAX} \{ \underset{\forall i \neq j \neq l \in \{1, 2, 3, \dots, 2^N\}}{MIN} [HD(B_i, B_j), HD(B_i, B_l), HD(B_j, B_l)] \} \quad (2)$$

Odległość Hamminga pomiędzy wektorami  $B_i, B_j$  jest równa  $HD(B_i, B_j) = S_{01}(B_i, B_j) + S_{10}(B_i, B_j)$  ( $S_{01}, S_{10}, S_{00}, S_{11}$  – miary zróżnicowania wektorów binarnych patrz [9]) i założmy, że jest ona maksymalna pomiędzy wszystkimi trzema wektorami, których szukamy. Chcąc otrzymać optymalny trzeci wektor  $B_l$  musimy zapewnić, aby odległość Hamminga pomiędzy tym wektorem a wektorami poprzednio otrzymanymi  $B_i$  i  $B_j$  była tak duża, jak to jest tylko możliwe. Aby to zapewnić wektor  $B_l$  musi być tak skonstruowany, aby był jednocześnie jak najbardziej różny od wektora  $B_i$  i wektora  $B_j$ .

Biorąc pod uwagę, iż wektory  $B_i = b_{i1}b_{i2} \dots b_{iN}$  i  $B_j = b_{j1}b_{j2} \dots b_{jN}$  posiadają  $S_{00}(B_i, B_j) + S_{11}(B_i, B_j)$  takich samych wartości na tych samych pozycjach ( $b_{ic} = b_{jc}$ ,  $c \in \{1, 2, \dots, N\}$ ), to wektor  $B_l$  powinien na tych samych pozycjach mieć wartości przeciwne do wartości będących w wektorach  $B_i$  i  $B_j$ . Można więc zapisać, iż  $HD(B_i, B_l)$  i  $HD(B_j, B_l)$  będą spełniały następujące nierówności:  $HD(B_i, B_l) \geq S_{00}(B_i, B_j) + S_{11}(B_i, B_j)$  i  $HD(B_j, B_l) \geq S_{00}(B_i, B_j) + S_{11}(B_i, B_j)$ . Wartość pozostałych  $S_{01}(B_i, B_j) + S_{10}(B_i, B_j)$  bitów w wektorze  $B_l$  powinna być ustalona tak, aby zapewniała maksymalną wartość  $HD(B_i, B_l)$  oraz  $HD(B_j, B_l)$ . W tym wypadku zwiększanie wartości  $HD(B_i, B_l)$  pociąga za sobą zmniejszanie  $HD(B_j, B_l)$ . Zatem aby osiągnąć maksimum obu tych wartości, należy dążyć do stanu w którym będą one sobie równe. Oznacza to, iż połowa z  $S_{01}(B_i, B_j) + S_{10}(B_i, B_j)$  różnych bitów wektorów  $B_i$  i  $B_j$  powinna zostać skopiowana na korespondujące pozycje do wektora  $B_l$  z wektora  $B_i$ , zaś druga połowa z wektora  $B_j$ . Zakładając, że wektor  $B_l$  jest wektorem wygenerowanym zgodnie z powyższą procedurą to gdy  $HD(B_i, B_j)$  jest parzyste, to jego odległości Hamminga  $HD(B_i, B_l)$  oraz  $HD(B_j, B_l)$  względem wektorów  $B_i$  i  $B_j$  można opisać wzorem (3).

$$HD(B_i, B_l) = HD(B_j, B_l) = S_{00}(B_i, B_j) + S_{11}(B_i, B_j) + [S_{01}(B_i, B_j) + S_{10}(B_i, B_j)] / 2 = N - HD(B_i, B_j) / 2, \quad (3)$$

W wypadku gdy  $HD(B_i, B_j)$  jest nieparzyste, to jego odległości Hamminga  $HD(B_i, B_l)$  oraz  $HD(B_j, B_l)$  względem wektorów  $B_i$  i  $B_j$  można opisać wzorami (4) i (5).

$$HD(B_i, B_l) = N - \lfloor HD(B_i, B_j) / 2 \rfloor \quad (4)$$

$$HD(B_j, B_l) = N - \lceil HD(B_i, B_j) / 2 \rceil \quad (5)$$

Problem wyznaczenia maksymalnej z minimalnych odległości Hamminga, zdefiniowany równaniem (2), może zostać sprowadzony do rozwiązania problemu opisanego równaniem (6).

$$MMHD(B_i, B_j, B_l) = \underset{HD(B_i, B_j)}{MAX} \{ \underset{HD(B_i, B_j)}{MIN} [HD(B_i, B_j), (N - \lceil HD(B_i, B_j)/2 \rceil), (N - \lfloor HD(B_i, B_j)/2 \rfloor)] \} \quad (6)$$

Rozwiązując powyższy problem i podstawiając  $X = HD(B_i, B_j)$  otrzymamy równanie:  $X = N - \lfloor X/2 \rfloor$ . Nieznana wartość  $X$ , która pozwoli otrzymać najmniejszą różnicę  $X - (N - \lfloor X/2 \rfloor) = 0$  lub  $1$  jest rozwiązaniem postawionego wcześniej problemu. Dla małych  $N$  optymalne wartości odległości Hamminga są zamieszczone w tab. 2

Tab.2. Optymalne odległości Hamminga dla małych  $N$

$N$	$HD(B_i, B_j)$	$HD(B_i, B_l)$	$HD(B_j, B_l)$
4	3	3	2
5	3	3	4
6	4	4	4
7	5	5	4
8	6	5	5
9	6	6	6
10	7	7	6
12	8	8	8

Opierając się na powyższych rozważaniach można zdefiniować Twierdzenie 2.

**Twierdzenie 2.** W przypadku trójprzebiegowej sesji testowej wykorzystującej do każdej iteracji inny wektor początkowy pamięci  $(B_i, B_j, B_l, i \neq j \neq l \in \{1, 2, \dots, 2^N\})$ , gdzie  $N$  rozmiar jednobitowej pamięci) oraz test generujący w zadanych komórkach tylko jedną klasę wzorców, najwyższa wydajność zostanie osiągnięta, gdy spełniona będzie równość wyrażona przez (7).

$$HD(B_i, B_j) = HD(B_i, B_l) = HD(B_j, B_l) \approx 2N/3 \quad (7)$$

Ta sama analiza przeprowadzona dla przypadku czteroprzebiegowej sesji testowej pozwala zdefiniować twierdzenie 3.

**Twierdzenie 3.** W czteroprzebiegowej sesji testowej wykorzystującej do każdej iteracji inny wektor początkowy pamięci  $(B_i, B_j, B_l, B_r, i \neq j \neq l \neq r \in \{1, 2, \dots, 2^N\})$ , gdzie  $N$  rozmiar jednobitowej pamięci) oraz test generujący w zadanych komórkach tylko jedną klasę wzorców, najwyższa wydajność zostanie osiągnięta, gdy spełniona będzie równość (8).

$$\begin{aligned} \text{HD}(B_i, B_j) &= \text{HD}(B_i, B_l) = \text{HD}(B_j, B_l) = \text{HD}(B_i, B_r) \\ &= \text{HD}(B_i, B_r) = \text{HD}(B_j, B_r) \approx 2N/3 \end{aligned} \quad (8)$$

## 5. ALGORYTM GENEROWANIA OPTYMALNYCH WEKTORÓW PAMIĘCI

Optymalnie dobrane początkowe wektory pamięci w testowaniu wieloprzebiegowym prowadzą do znacznego zwiększenia wydajności procesu testowania w przypadku uszkodzeń uwarunkowanych zawartością. Jeden z przykładów optymalnie dobranych wektorów pamięci w wypadku czteroprzebiegowej sesji testowej i  $N = 9$  może mieć postać następującą:

$$\begin{aligned} B_1 &= b_{11}b_{12}b_{13}b_{14}b_{15}b_{16}b_{17}b_{18}b_{19} = 000000000 \\ B_2 &= b_{21}b_{22}b_{23}b_{24}b_{25}b_{26}b_{27}b_{28}b_{29} = 111111000 \\ B_3 &= b_{31}b_{32}b_{33}b_{34}b_{35}b_{36}b_{37}b_{38}b_{39} = 111000111 \\ B_4 &= b_{41}b_{42}b_{43}b_{44}b_{45}b_{46}b_{47}b_{48}b_{49} = 000111111. \end{aligned}$$

W przedstawionym przykładzie pierwszy wektor pamięci złożony jest z samych wartości 0. Należy jednak podkreślić, iż początkowy wektor pamięci może przyjąć postać dowolną, np.:  $B_1 = 010011011$ . Zaznaczyć należy również, iż w przypadku generowania zbioru wektorów dla czteroprzebiegowej sesji testowej, drugi wektor można wyznaczyć poprzez negację dowolnych  $2N/3$  bitów pierwszego wektora. Oznaczmy przez P1 zbiór adresów połowy z  $2N/3$  zanegowanych bitów, zaś przez P2 – zbiór adresów drugiej połowy. Obie części mają tę samą liczbę bitów równą  $N/3$ . P3 niech oznacza pozostałe bity. Jako algorytm podziału wektora  $B_1 = b_{11}b_{12}...b_{1N}$ , gdzie  $b_{1c} \in \{0, 1\}$ ,  $\forall c \in \{1, 2, \dots, N\}$  na poszczególne części może zostać przyjęty dowolny algorytm będący w stanie wygenerować wszystkie wartości  $c$  (adresy). Jako dobre rozwiązanie do selekcji poszczególnych bitów wydaje się rozwiązanie oparte o sekwencję generowaną przez rejestr LFSR. Przykładowo, dla  $N = 9$  wygenerowane wartości  $c$  określające kolejne bity wektora  $B_1 = b_{11}b_{12}b_{13}b_{14}b_{15}b_{16}b_{17}b_{18}b_{19}$  mogą przybrać postać  $c = \{2, 7, 1, 8, 3, 5, 9, 4, 6\}$ . Wtedy, P1 =  $\{2, 7, 1\}$ , P2 =  $\{8, 3, 5\}$  zaś P3 =  $\{9, 4, 6\}$ . W oparciu o powyższe założenia zdefiniowany poniżej algorytm przedstawia sekwencję kroków pozwalających wygenerować optymalne wektory w wypadku czteroprzebiegowej sesji testowej opartej o test klasy MATS+.

**Algorytm 1:** Generowanie optymalnych wektorów pamięci  $B_1, B_2, B_3$  i  $B_4$

1. Wyznaczenie zbiorów bitów P1, P2 i P3
2. Wygenerowanie wektora  $B_2$ , poprzez inwersję bitów ze zbioru P1, P2 wektora  $B_1$
3. Wygenerowanie wektora  $B_3$  poprzez inwersję bitów ze zbiorów P3, P1 wektora  $B_2$
4. Wygenerowanie wektora  $B_4$  poprzez inwersję bitów ze zbiorów P1, P2 wektora  $B_2$ .

Zaproponowany algorytm oparty o procedurę sekwencyjnego generowania pozycji bitów  $c$ , które podzielone zostały na obszary P1, P2, P3, w prosty sposób pozwala wygenerować optymalne wektory dla czteroprzebiegowej sesji testowej opartej o test klasy MATS+. Wygenerowanie wektora  $B_2$  odbywa się na bazie wektora pamięci  $B_1$  poprzez inwersję bitów należących do grupy P1 i P2. Wektor  $B_3$  otrzymujemy na bazie wektora  $B_2$  poprzez inwersję bitów należących do grupy P3 i P1, zaś wektor  $B_4$  otrzymujemy na bazie wektora  $B_3$  poprzez inwersję bitów należących do grupy P1 i P2.

## 6. WNIOSKI

W artykule rozpatrzony został problem generowania optymalnych wektorów początkowych pamięci w wypadku wieloprzebiegowego jej testowania. Testowanie wieloprzebiegowe używane jest do efektywnej detekcji uszkodzeń uwarunkowanych zawartością. W artykule przedstawione zostały warunki, które muszą być spełnione przez wektory optymalne dla przypadku sesji dwu, trzy i czteroprzebiegowych testów krokowych generujących tylko jeden wzorzec w wybranych komórkach pamięci. Dodatkowo w artykule zaproponowany został algorytm umożliwiający w sposób efektywny generowanie optymalnych wektorów dla przypadku sesji czteroprzebiegowych. Otwartym problemem pozostaje wyznaczenie optymalnej odległości Hamminga w wypadku sesji testowych cechujących się większą liczbą iteracji niż cztery. Z uwagi na powyższe otwarty jest również problem generowania wektorów pamięci charakteryzujących się taką odległością.

## 7. BIBLIOGRAFIA

- [1] Marinissen E. J., Prince B., Keitel-Schulz D., Zorian Y.. Challenges in embedded memory design and test. In DATE '05: Proceedings of the conference on Design, Automation and Test in Europe, pages 722–727, 2005.
- [2] van de Goor A. J.. Testing Semiconductor Memories: Theory and Practice. John Wiley & Sons, Chichester, England, 1991
- [3] Cockburn B. F. Deterministic tests for detecting scrambled pattern-sensitive faults in RAMs. In MTDT '95: Proceedings of the IEEE International Workshop on Memory Technology, Design and Testing, pages 117–122, Washington, DC, USA, 1995. IEEE Computer Society
- [4] Franklin M., Saluja K. K.. Testing reconfigured RAM's and scrambled address RAM's for pattern sensitive faults. IEEE Transactions on CAD of Integrated Circuits and Systems, 15(9):1081–1087, 1996.
- [5] Karpovsky M. G., van de Goor A. J., Yarmolik V. N.. Pseudo-exhaustive word-oriented DRAM testing. In EDTC '95: Proceedings of the 1995 European conference on Design and Test, page 126, Washington, DC, USA, 1995. IEEE Computer Society.
- [6] Nicolaidis M. «Transparent BIST for RAMs», Proc. IEEE Int. Test Conf., Baltimore, MD, Oct. 1992, pp. 598-607
- [7] Sokol B., Yarmolik S. V. Address sequences for march tests to detect pattern sensitive faults. In DELTA '06: Proceedings of the Third IEEE International Workshop on Electronic Design, Test and Applications, pages 354–360, Kuala Lumpur, Malaysia, January 17-19 2006. IEEE Computer Society
- [8] Yarmolik S. V., Mrozek I. Multi background memory testing. In MIXDES 07: Proceedings of the 14th International Conference Mixed design of integrated circuits and systems, pages 511–516, Ciechocinek, Poland, June 21-23 2007. IEEE Computer Society.
- [9] Mrozek I., Yarmolik V., Optimal backgrounds selection for multi run memory testing, Design and diagnostics of electronic circuits and systems : 11th International Workshop IEEE : DDECS 2008, Bratislava (Słowacja), 2008, pp. 332-338