

Ireneusz MROZEK¹

TRANSFORMACJA TRANSPARENTNYCH TESTÓW KROKOWYCH W TESTY SYMETRYCZNE

Pamięci półprzewodnikowe wykorzystywane są obecnie we wszystkich systemach cyfrowych w tym w systemach o znaczeniu krytycznym, w których proces testowania poszczególnych podzespołów musi odbywać się periodycznie podczas normalnej pracy całego systemu. W wypadku pamięci RAM i testowania periodycznego bardzo ważną rolę odgrywają transparentne testy krokowe. Jedną z najbardziej interesujących technik pozwalających na implementację testów transparentnych jest technika oparta na testach symetrycznych. W artykule przedstawiona zostanie metoda transformacji dowolnego niesymetrycznego testu krokowego w postać symetryczną.

TRANSFORMATION OF TRANSPARENT MARCH ALGORITHMS INTO SYMMETRIC TESTS

High-density memories are often used in safety-critical microelectronic systems. Transparent March algorithms allow realizing periodic testing of memory while preserving its contents. One of the most interesting technique, which allows to implement the transparent memory testing is based on symmetry of march tests. This paper presents the technique which allows to transform any asymmetric march test to the symmetric one.

1. WSTĘP

Kluczową rolę w technikach testowania pamięci odgrywają testy krokowe (ang. *march tests*)[1-3]. Są one obecnie najczęściej stosowanymi testami w procesie testowania pamięci. Test krokowy 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 –

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

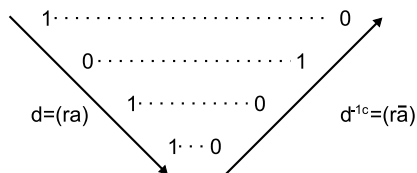
odczytaj wartość z komórki pamięci (wartość spodziewana jest równa x). Cały test typu March ograniczony jest poprzez parę nawiasów klamrowych { ... }, podczas gdy faza testu krokowego ograniczona jest parą nawiasów okrągłych (...). Jako przykład rozpatrzmy test MATS+ [1]:

$$\{ \underset{M0}{\Downarrow}(w0); \underset{M1}{\Uparrow}(r0, w1); \underset{M2}{\Downarrow}(r1, w0) \}.$$

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. Przy standardowej implementacji testu krokowego o uszkodzeniu pamięci informuje fakt odczytania innej wartości niż wartość spodziewana. Największymi zaletami testów krokowych są: bardzo wysoka (100 %) wykrywalność uszkodzeń związanych z jedną i dwoma komórkami pamięci jak również liniowa złożoność rzędu N (N – rozmiar testowanej pamięci). Ponadto zgodnie ze schematem zaproponowanym w [4] testy te w sposób prosty można przekształcić na postać transparentną. Pozwala to wykorzystywać je do cyklicznego sprawdzania poprawności działania pamięci bez niszczenia jej zawartości. Tradycyjna technika transparentnego testowania pamięci oparta jest na liniowej analizie sygnatury. Poprzez sygnaturę rozumiana jest wartość zależna od zawartości pamięci (suma kontrolna pamięci liczona wg. określonego algorytmu). W tym wypadku rezultat testu jest wynikiem porównania wartości sygnatury pamięci z momentu poprzedzającego rozpoczęcie testu pamięci z wartością sygnatury pamięci wyliczonej po zakończeniu testu (lub wyliczanej w trakcie jego trwania). Takie podejście do testowania pamięci wymaga jednak przed rozpoczęciem testu wyliczenia sygnatury odniesienia. Powoduje to przedłużeniu procesu testowania o około 50% [4]. Dlatego zaproponowane zostały rozwiązania pozwalające skrócić ten proces. Jednym z najbardziej znanych podejść do problemu jest zastosowanie techniki testowania opartej o testy symetryczne [5].

2. SYMETRYCZNE TESTY PAMIĘCI

W większości transparentnych krokowych algorytmów testowania pamięci generowany wyjściowy ciąg danych cechuje się wysokim stopniem symetrii. Jako przykład rozpatrzmy transparentną odmianę algorytmu MATS+ - $\{ \underset{M0}{\Uparrow}(ra, w\bar{a}); \underset{M1}{\Downarrow}(r\bar{a}, wa) \}$ - działającą na pamięci o pojemności 4 bitów i zawartości $\{1,0,1,1\}$. Zgodnie z pierwszą fazą testu dane wyjściowe przesyłane do rejestru (licznika) sygnatury będą miały następującą postać: $d=(1,0,1,1)$. Podczas drugiej fazy $\underset{M1}{\Downarrow}(r\bar{a}, wa)$ dane wyjściowe pojawiają się w kolejności odwrotnej oraz ich wartości będą negacją logiczną wartości występujących w strumieniu danych wyjściowych pierwszej fazy. I tak dane te mają postać następującą: $d^*=(0,0,1,0)$. Sumaryczny ciąg wyjściowy z testu przyjmie zatem postać: $dd^* = (1,0,1,1|0,0,1,0)$. Można zauważyć symetrię drugiej części ciągu w stosunku do pierwszej, polegającą na lustrzanym odbiciu danych z pierwszej części i zanegowaniu ich wartości. Graficznie przedstawia to rys. 1.



Rys.1. Symetria w teście MATS+

Połączenie własności symetrii ciągu wyjściowego z odpowiednim sposobem liczenia wartości sygnatury pozwala na całkowite pominięcie fazy wstępnej testu (liczenie sygnatury odniesienia) - niezbędnej w tradycyjnych realizacjach testów transparentnych. Symetryczny ciąg danych wyjściowych i zmiana, w drugiej części testu wielomianu używanego w celu otrzymania sygnatury, na odpowiedni wielomian odwrotny (*ang. reciprocal polynomial*), pozwala precyzyjnie określić wartość sygnatury odniesienia bez potrzeby realizacji fazy wstępnej testu, która była niezbędna dotychczas [5]. Należy również dodać, iż wartość ta jest niezależna od zawartości i rozmiaru testowanej pamięci. Analizując istniejące testy krokowe można stwierdzić, iż w testach tych pojawiają się dwa typy symetrii. Zakładając, iż pierwszą część ciągu generowanego przez test można zapisać jako $d = (d_0, d_1, \dots, d_{n-1}) \in \{0,1\}^n$ wtedy mówimy, iż:

- ciąg danych wyjściowych charakteryzuje się symetrią TYPU 1, jeżeli można go zapisać jako dd^1 ;
- ciąg danych wyjściowych charakteryzuje się symetrią TYPU 2, jeżeli można go zapisać jako dd^{1c} ,

gdzie, $d^1 := (d_{n-1}, \dots, d_0) \in \{0,1\}^n$, $d^{1c} := (d_{n-1}^c, \dots, d_0^c) \in \{0,1\}^n$, a wartość d_k^c jest wartością komplementarną do wartości d_k .

W [5] zostało udowodnione twierdzenie mówiące, że $S(d^1 S(d,s,h)-1, h^1) = s^1$, gdzie d i d^1 jest odpowiednio pierwszą i drugą częścią danych kierowanych do analizatora sygnatury, a $b = S(d,s,h)$ oznacza wartość sygnatury ciągu d , wyliczoną z użyciem wielomianu charakterystycznego h przy początkowym stanie rejestru sygnatury równym s . Powyższe twierdzenie oznacza, że dla odwróconego ciągu danych d^1 analizowanego przez analizator sygnatury ze stanem początkowym b^1 (odwrócona sygnatura ciągu d) i wielomianem charakterystycznym h^1 (wielomian odwrotny w stosunku do wielomianu h), otrzymuje się wartość sygnatury s^1 (odwróconą wartość początkową rejestru sygnatury). Tak więc dla testu, którego ciąg danych wyjściowych charakteryzuje się symetrią TYPU 1 – dd^1 – nie ma konieczności liczenia sygnatury odniesienia, gdyż dla danych nie zawierających błędów jest ona znana na podstawie wartości początkowej rejestru sygnatury. Podobne twierdzenie zostało sformułowane i udowodnione dla symetrii TYPU 2. Również zostało pokazane, iż wartość końcową rejestru sygnatury (w przypadku analizy danych postaci dd^{1c}), można określić na podstawie wartości początkowej tego rejestru. Widać więc, iż zastosowanie testów symetrycznych w testowaniu transparentnym może znacznie uprościć proces testowania pamięci RAM.

3. PRZEKSZTAŁCENIE TESTÓW NIESYMETRYCZNYCH W TESTY SYMETRYCZNE

Technika testów symetrycznych pozwala znacznie skrócić czas transparentnego testowania pamięci. Jednak nie wszystkie testy krokowe charakteryzują się taką konstrukcją, która umożliwi generowanie ciągu wyjściowego mającego cechy symetrii. Porównanie wybranych testów krokowych pod kątem symetrii generowanych przez nie danych wyjściowych zamieszczone jest w tab. 1.

Tab. 1. Symetria w testach krokowych

<i>Nazwa testu</i>	<i>Typ symetrii</i>
<i>MATS</i>	<i>TYP 2</i>
<i>MATS+</i>	<i>TYP 2</i>
<i>MATS++</i>	<i>BRAK</i>
<i>MARCH X</i>	<i>BRAK</i>
<i>MARCH Y</i>	<i>BRAK</i>
<i>MARCH C-</i>	<i>BRAK</i>
<i>MARCH C</i>	<i>TYP 1</i>
<i>MARCH A</i>	<i>TYP 1</i>
<i>MARCH B</i>	<i>TYP 1</i>

W celu realizacji testu z użyciem techniki symetrycznych testów pamięci należy sprawdzić, czy dany test generuje ciąg wyjściowy zgodny z odpowiednim typem symetrii. Jeżeli ciąg wyjściowy nie spełnia tych warunków, test należy tak przekształcić, żeby wyjściowy ciąg danych był odpowiedni. Rozpatrując symetrię TYPU 2 należy zadany test przekształcić tak, by dane wyjściowe można było zapisać w postaci dd^{lc} . Można to uzyskać poprzez dodawanie poleceń odczytu, gdyż zgodnie z własnością 1 nie pogorszy to jakości testu.

Własność 1: *Pokrycie uszkodzeń testu krokowego nie zmniejszy się po dodaniu do dowolnej jego fazy, w dowolnym miejscu, dowolnej liczby poleceń odczytu oraz po dodaniu do testu nowych faz składających się wyłącznie z poleceń odczytu.*

Uzasadnienie tej własności opiera się na charakterystycznych cechach stosowanych modeli uszkodzeń. Główną rolę w technice testowania pamięci nadal odgrywają tradycyjne modele uszkodzeń [1]. W modelach tych uszkodzenie nie ulega aktywacji podczas wykonywania polecenia odczytu zawartości komórki pamięci, dlatego też nie może ono powodować maskowania błędów. Zatem dodanie dowolnej liczby poleceń odczytu nie wpływa na wydajność testu (w sensie pokrycia uszkodzeń). W wypadku uszkodzeń aktywowanych poleceniem odczytu (np. uszkodzenie RDF) każdorazowo do problemu należy podchodzić

odmiennie. W większości wypadków dodatkowe polecenia odczytu spowodują zwiększenie pokrycia uszkodzeń w zadanym teście, a nie jego zmniejszenie.

Niestety, dodatkowe polecenia odczytu zwiększają złożoność testu. Jednak, jak wynika z proponowanego twierdzenia 1, nie będzie ona większa niż łączna złożoność testu w klasycznej postaci (złożoność liczenia sygnatury i złożoność samego testu).

Twierdzenie 1: *Złożoność transparentnego testu krokowego, przekształconego do postaci transparentnej poprzez dodanie poleceń odczytu, jest mniejsza bądź równa, sumarycznej złożoności testu realizowanego tradycyjną techniką transparentnego testowania pamięci, na którą składa się złożoność algorytmu obliczania sygnatury odniesienia oraz złożoność samego testu.*

Aby udowodnić powyższe twierdzenie, rozpatrzony zostanie skrajny przypadek testu składającego się z n $P_0P_1\dots P_{n-1}$ faz, z identycznym kierunkiem adresowania. Jeżeli w fazie P_i będzie k_{iw} poleceń zapisu oraz k_{ir} poleceń odczytu, to złożoność całkowita testu (z_{tr}), realizowanego z użyciem tradycyjnej techniki transparentnego testowania pamięci, na którą składa się złożoność algorytmu obliczania sygnatury i złożoność samego testu, jest równa:

$$z_{tr} = \sum_{i=0}^{n-1} k_{ir} + \sum_{i=0}^{n-1} (k_{ir} + k_{iw}).$$

Aby uzyskać postać symetryczną testu, można poprzedzić go n dodatkowymi fazami $P_{n-1}^* \dots P_1^* P_0^*$, których kierunek adresowania powinien być przeciwny do kierunku adresowania w fazach testu oryginalnego. Zakładając, że każda dodana faza będzie miała tyle samo poleceń odczytu, ile było w teście oryginalnym, przyjmie on następującą postać: $P_{n-1}^* \dots P_1^* P_0^* P_0 P_1 \dots P_{n-1}$. Złożoność takiego testu (z_{nw}) będzie więc równa:

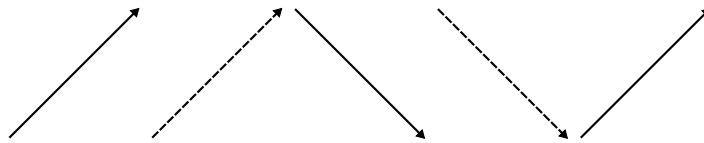
$$z_{nw} = \sum_{i=0}^{n-1} k_{ir} + \sum_{i=0}^{n-1} (k_{ir} + k_{iw}) = z_{tr}.$$

Zatem po przekształceniu testu do postaci transparentnej, poprzez dodanie poleceń odczytu, jego złożoność będzie nie większa niż złożoność oryginalnego testu realizowanego w technice tradycyjnej. ■

W rzeczywistości przekształcenie testów do postaci symetrycznej będzie wymagało zazwyczaj dodania tylko niewielu dodatkowych poleceń odczytu. W związku z tym, złożoność takiego testu będzie zwykle dużo mniejsza od złożoności testu realizowanego techniką tradycyjną. Jako przykład proponowanego podejścia rozważone zostanie przekształcenie do postaci symetrycznej testu March C-. Test March C- w wersji nietransparentnej ma postać:

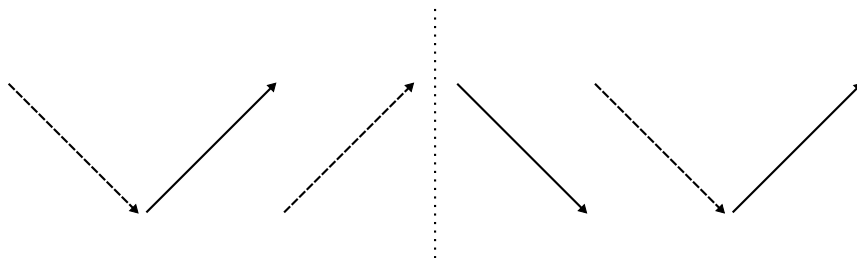
$$\uparrow\downarrow(w0); \uparrow\uparrow(r0,w1); \uparrow\uparrow(r1,w0); \downarrow\downarrow(r0,w1); \downarrow\downarrow(r1,w0); \uparrow\downarrow(r0).$$

Złożoności transparentnej wersji bazowej $\hat{\uparrow}(ra, w\bar{a}); \hat{\uparrow}(r\bar{a}, wa); \Downarrow(ra, w\bar{a}); \Downarrow(r\bar{a}, wa); \hat{\uparrow}(ra)$ równa jest $9N$, zaś procedury wyliczenia sygnatury odniesienia $\hat{\uparrow}(ra); \hat{\uparrow}(r\bar{a}); \Downarrow(ra); \Downarrow(r\bar{a}); \hat{\uparrow}(ra) - 5N$. Zatem finalna złożoność transparentnej procedury testującej opartej o test March C- przy podejściu tradycyjnym jest równa $14N$ [6]. Z analizy powyższego testu wynika, że generowany przez niego wyjściowy ciąg danych nie spełnia warunków symetrii. Graficzny schemat testu przedstawiony jest na rys.2.



Rys.2. Graficzny schemat testu March C-

Każde polecenie odczytu występujące w teście oznaczone jest strzałką. Kierunek strzałki oznacza kierunek adresowania fazy, w której dane polecenie się znajduje, strzałka z linią ciągłą oznacza polecenie ra , a strzałka z linią przerywaną oznacza polecenie $r\bar{a}$. Transformacja testu March C- do postaci symetrycznej (TYP 2) polega na dodaniu poleceń odczytu w taki sposób, aby postać graficzna testu miała charakter odwrotnej symetrii. Należy zatem zbudować test, którego schemat graficzny będzie zgodny ze schematem z rys. 3.



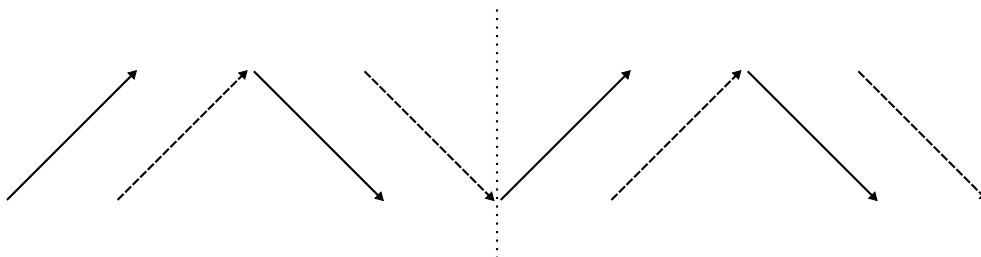
Rys.3. Schemat graficzny testu March C- po transformacji do postaci symetrycznej (wer. 1)

Aby to uzyskać, wystarczy dodać jako pierwszą fazę testu, fazę odczytu wartości odwrotnej. W wyniku przekształcenia test będzie miał następującą postać:

$$\{\Downarrow(r^*\bar{a}); \hat{\uparrow}(ra, w\bar{a}); \hat{\uparrow}(r\bar{a}, wa); \Downarrow(ra, w\bar{a}); \Downarrow(r\bar{a}, wa); \hat{\uparrow}(ra)\}.$$

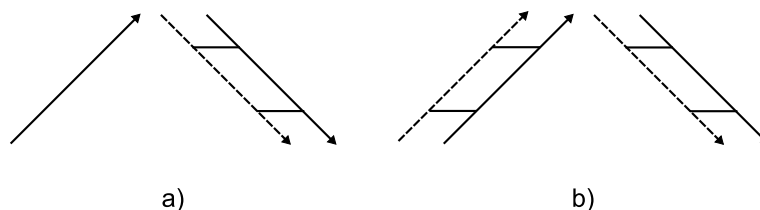
$$P_2 \quad P_1 \quad P_0 \quad P_0^* \quad P_1^* \quad P_2^*$$

Polecenie odczytu oznaczone jako (r^*a) polega na odczytaniu z komórki pamięci wartości bieżącej i skierowaniu do analizatora sygnatury wartości przeciwnej. Uzyskany wyjściowy ciąg danych będzie cechował się symetrią TYPU 2. Dane z faz $P_2P_1P_0$ utworzą ciąg d , zaś dane z faz $P_0^*P_1^*P_2^*$ ciąg $d^{l.c}$. Złożoność testu wyniesie $10N$, czyli będzie o około 30% mniejsza od złożoności testu realizowanego techniką tradycyjną. Należy zauważyć, iż test może zostać doprowadzony do postaci symetrycznej na wiele sposobów. Schemat graficzny innej symetrycznej postaci testu March C- przedstawia rys. 4.



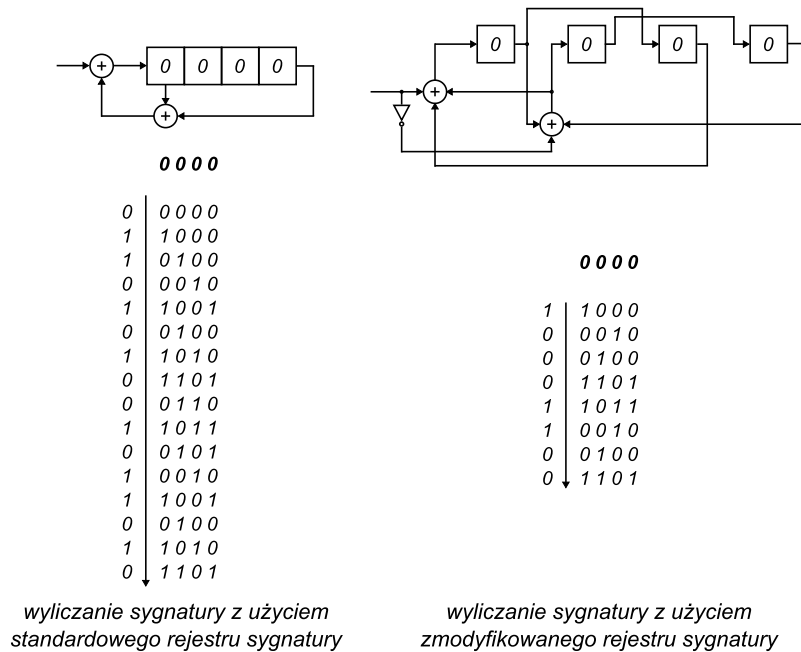
Rys.4. Test March C- po transformacji do postaci symetrycznej (wer. 2)

W wypadku konieczności przekształcania testu do postaci symetrycznej poprzez dodanie odpowiedniej liczby poleceń odczytu, istnieje niekiedy możliwość optymalizacji implementacji dodanych poleceń odczytu. Zostanie to prześledzone na przykładzie testu MATS++, który w wersji transparentnej jest zapisywany jako: $\{\uparrow(r\bar{a}, w\bar{a}); \downarrow(r\bar{a}, w\bar{a}, ra)\}$. Schemat graficzny testu przedstawia rys. 5a, gdzie strzałki połączone linią poziomą oznaczają polecenia odczytu z tej samej fazy. Test MATS++ w standardowej postaci nie jest testem symetrycznym. Składa się on z dwóch faz - pierwsza zawiera jedno polecenie odczytu, zaś druga - dwa. Przekształcenie testu do postaci symetrycznej można zrealizować dodając polecenie odczytu do pierwszej fazy testu (patrz rys. 4b).



Rys.5. Schemat graficzny testu MATS++ a) w wersji oryginalnej b) w wersji symetrycznej

Zmodyfikowany test MATS++ do postaci symetrycznej zapisać można jako: $\{\uparrow(r^*\bar{a}, ra, w\bar{a}); \downarrow(r\bar{a}, w\bar{a}, ra)\}$. Złożoność tego testu wynosi $6N$, więc jest równa złożoności testu w postaci nietransparentnej. Występujące w przekształconym teście dwa polecenia odczytu obok siebie, można jednak wykorzystać w implementacji sprzętowej i zmniejszyć złożoność testu. Ponieważ między dwoma poleceniami odczytu nie występuje polecenie zapisu, fizyczne odczytanie zawartości komórki może być tylko jednorazowe. Kolejna wartość może zostać ustalona na podstawie wartości już odczytanej, co jest możliwe do osiągnięcia, np. poprzez rozbudowę analizatora sygnatury. Fakt ten zobrazowany jest na rys. 6.



Rys.6. Obliczenie wartości sygnatury dla pierwszej fazy symetrycznej wersji testu MATS++

W przykładzie z rys. 6 analizator sygnatury SA jest zrealizowany w oparciu o wielomian $h(x)=1+x+x^4$, a testowaniu podlega pamięć z zawartością 10001100. Zgodnie z symetryczną postacią testu MATS++, pierwsza jego faza ($r^*ā,ra,wā$) do rejestru sygnatury skieruje ciąg danych $d=0110101001011010$. Aby skrócić czas testowania, fizycznie można realizować fazę w postaci $(ra,wā)$ zaś wynik dodatkowego polecenia odczytu generować w analizatorze sygnatury. Wtedy ciąg kierowany do rejestru sygnatury będzie miał postać $d=10001100$. Ze schematu przedstawionego na rys. 6 wynika, że modyfikacja rejestru sygnatury polega na dodaniu jednego negatora, jak również elementów umożliwiających otrzymanie sumy modulo 2. Jeżeli w danej fazie występują dwa jednakowe polecenia odczytu, ze schematu należy usunąć negator. Podobny schemat, jak przedstawiony w powyższym przykładzie, można utworzyć dla trzech i większej liczby poleceń odczytu występujących obok siebie. Umożliwia to, przy niewielkim narzucie sprzętowym, znaczne skrócenie czasu testu.

4. PODSUMOWANIE

Zaprezentowana technika przekształcania testów niesymetrycznych w testy symetryczne polega na dodawaniu w odpowiednich miejscach testu poleceń odczytu. Prowadzi to jednak do zwiększenia złożoności testu, a w konsekwencji wydłuża czas testu. Minimalizacji skutków czasowych spowodowanych dodatkowymi poleceniami odczytu można dokonać poprzez odpowiednią implementację rejestru sygnatury. Innym rozwiązaniem umożliwiającym uniknięcie dodatkowego narzutu jest zastosowanie techniki

opartej na wielokrotnych rejestrach sygnatury, w których wykorzystywana jest symetria faz testu (symetrie cząstkowe). Przykłady takich rozwiązań można znaleźć m.in. w [7,8].

6. BIBLIOGRAFIA

- [1] Van de Goor A. J.: *Testing Semiconductor Memories, Theory and Practice*. Chichester, John Wiley & Sons, 1991.
- [2] Al-Harbi, S.M. ; Noor, F. ; Al-Turjman, F.M. : March DSS: A New Diagnostic March Test for All Memory Simple Static Faults, *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, Vol. 26 , Issue 9, pp. 1713 – 1720, September 2007
- [3] Sosnowski J.: *Testowanie i niezawodność systemów komputerowych*. Akademicka Oficyna Wydawnicza EXIT, 2005.
- [4] Nicolaidis M.: Transparent BIST for RAMs. In *Proceedings of the IEEE International Test Conference*, Baltimore, MD, pp. 598-607, 1992.
- [5] 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.
- [6] Nicolaidis M.: Theory of transparent BIST for RAMs. *IEEE Transaction of Computers*, 45(10):1141–1156, October 1996.
- [7] Mrozek I.: Symetryczne, niedestrukcyjne techniki identyfikacji i rozpoznawania błędów działania pamięci. Materiały konferencyjne „Inżynieria wiedzy i systemy ekspertowe”, s. 399-406, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, 2000.
- [8] Mrozek I.: Wykrywanie i lokalizacja uszkodzeń pamięci RAM oparte o transparentne testy pamięci, rozprawa doktorska , Politechnika Białostocka, Białystok 2004.