

Rozdział 4

Dwuprzebiegowe testy krokowe z regularnym indeksem inkrementacji

Ireneusz Mrozek
Wydział Informatyki, Politechnika Białostocka

Streszczenie: Konwencjonalne testy krokowe (ang. *march test*) pamięci RAM wyróżniają się szczególnie wysoką efektywnością w przypadku uszkodzeń prostych, takich jak: uszkodzenie sklejeniuowe (ang. *Stuck-at fault* – SAF), uszkodzenie przejścia (ang. *Transition fault* – TF) czy uszkodzenie sprzężeniowe (ang. *Coupling fault* – CF). Należy jednak zaznaczyć, iż sesje testowe złożone z pojedynczego wykonania testu krokowego, są niewystarczające w przypadku uszkodzeń złożonych takich, jak uszkodzenia uwarunkowane zawartością sąsiadów (ang. *Pattern Sensitive Faults* – PSF). Jedną z technik zwiększających pokrycie uszkodzeń PSF jest technika sesji wieloprzebiegowych, której ideą jest wielokrotne wykonanie transparentnego testu krokowego przy różnych warunkach początkowych. Rozwiązanie to jest szczególnie interesujące w przypadku wbudowanych testów automatycznych (ang. *Built-in self-test* – BIST). Podejście oparte na testach transparentnych posiada niepodważalną zaletę, w stosunku do testowania tradycyjnego. Po zakończeniu procesu testowania, zawartość pamięci RAM jest taka sama jak przed rozpoczęciem testu. Nie istnieje w tym przypadku potrzeba zapisywania zawartości testowanej pamięci przed rozpoczęciem sesji testowej, a następnie jej odtwarzania po zakończeniu procesu testowania. Jest to jedna z przyczyn, dla których technika testowania pamięci RAM, oparta na wieloprzebiegowych transparentnych testach krokowych, jest szeroko stosowana do cyklicznych testów pamięci systemów o znaczeniu krytycznym (elektronika medyczna, sterowanie kolejami, awionika, telekomunikacja i innych). W wielu przypadkach dostępny czas na realizację takiej sesji testowej jest limitowany. Biorąc pod uwagę powyższe ograniczenie w pracy uwaga skupiona zostanie nad krótkimi, dwuprzebiegowymi sesjami testowymi wykorzystującymi zmienne sekwencje adresowe generowane na podstawie regularnego indeksu inkrementacji. W pracy dokonano dogłębnej analizy warunków, które muszą spełniać sekwencje adresowe, aby uzyskać jak największe pokrycie uszkodzeń. Określono optymalną wartość indeksu inkrementacji prowadzącą do uzyskania maksymalnego pokrycia uszkodzeń. Wyniki prac wskazują, iż sesje testowe wykorzystujące tak generowane sekwencje adresowe dają wyższe pokrycie uszkodzeń niż ma to miejsce w przypadku sekwencji pseudolosowych. Jako uzupełnienie zaproponowano rozwiązanie o niskim narzucie sprzętowym, umożliwiające implementację generatora sekwencji adresowych opartych na optymalnym indeksie inkrementacji.

Słowa kluczowe: RAM, BIST, PSF, testowanie pamięci, testy transparentne

Afiliacja: Publikacja zrealizowana w ramach pracy statutowej WZ/WI-IIT/2/2020.

Wprowadzenie

Pamięć półprzewodnikowa jest kluczowym elementem szeroko rozumianych współczesnych systemów i układów elektronicznych. Jej bezawaryjne działanie ma fundamentalny wpływ na zachowanie całego systemu. Procentowy udział podłoża krzemowego przeznaczonego na różne kategorie pamięci w systemach typu SoC (ang. *systems-on-chip*) jest obecnie na poziomie 90% [2]. Produkowana pamięć musi spełniać bardzo wysokie wymagania związane z miarą jakości (od 50 ppm w przypadku standardowych komputerów do poniżej 10 ppm w przypadku układów stosowanych w systemach o wysokiej wiarygodności działania), [2]. Obecna technologia umożliwia produkcję układów pamięci wysokiej gęstości o niespotykanych dotychczas pojemnościach. Jednocześnie wzrost upakowania układów i tym samym zmniejszenie fizycznych odległości pomiędzy poszczególnymi jego elementami sprzyja niezamierzonym interakcjom pomiędzy sąsiadującymi komórkami pamięci. Fizyczną przyczyną tych interakcji są z reguły prądy upływu [8]. W konsekwencji, wzajemna interakcja międzykomórkowa zależy w znacznej mierze od wartości logicznych występujących w tych komórkach, czego skutkiem są błędy występujące tylko w kontekście specyficznych wartości (wzorców) tam zapisanych. Wpływa to znacząco na wzrost stopnia złożoności procesów testowania pamięci. Należy podkreślić, że w obecnych pamięciach DRAM błędy będące wynikiem wzajemnego oddziaływania komórek na siebie są jednymi z najczęściej zgłaszanych podczas procesów testowania [5]. Mimo, że waga i mechanizm powstawania tych uszkodzeń znane są od dawna, to wciąż nie zostały opracowane metody umożliwiające dokładną i efektywną analizę pamięci pod tym kątem [5]. Fakt ten sprawia, iż nadal istnieje zapotrzebowanie rozwijania technik i algorytmów w tym obszarze.

Na potrzeby testowania pamięci zdefiniowanych zostało wiele modeli uszkodzeń (ang. *fault models*). Definicje podstawowych z nich można znaleźć w pracach Thatte i Abrahama w [6]. Liczna grupa modeli została zaproponowana przez Van de Goora w [8]: uszkodzenia dekodera adresu (ang. *address decoder faults* – AF), uszkodzenia sklejenkowe (ang. *stuck-at faults* – SAF), uszkodzenia sprzężeniowe (ang. *coupling faults* – CF), uszkodzenia przejścia (ang. *transition faults* – TF) czy uszkodzenia uwarunkowane zawartością (ang. *pattern sensitive faults* – PSF), w tym uszkodzenia uwarunkowane zawartością sąsiadów (ang. *neighborhood pattern sensitive faults* – NPSF). Rozwinięcia powyższych modeli dla różnych typów pamięci można znaleźć m.in. w [1, 3, 4, 5, 7]. Przyjęło się również dzielić uszkodzenia pamięci ze względu na liczbę komórek, które są powiązane danym uszkodzeniem. Dotyczy to uszkodzeń prostych związanych z pojedynczymi komórkami pamięci (ang. *one cell faults*) i uszkodzeń złożonych, wiążących zależnościami wiele komórek pamięci (ang. *multi-cell faults*).

Ze względu na szybko rosnącą pojemność i gęstość upakowania układów pamięci, szczególnego znaczenia nabierają modele prowadzące do opracowania testów umożliwiających efektywne wykrywanie uszkodzeń, których źródłem jest wzajemna interakcja pomiędzy poszczególnymi komórkami pamięci. Defekty, których rezultatem

są interakcje między komórkami pamięci najczęściej są analizowane z wykorzystaniem dwóch dobrze udokumentowanych w literaturze modeli uszkodzeń: uszkodzenia sprzężeniowego CF i uszkodzenia uwarunkowanego zawartością PSF [8]. Uszkodzenie CF opisuje interakcję dwóch dowolnych komórek pamięci [8]. O uszkodzeniu tym mówi się wówczas, gdy zmiana wartości (lub określona wartość) w komórce określonej mianem agresora wpływa na wartość w komórce określonej mianem ofiary.

Uogólnionym modelem umożliwiającym analizę oddziaływania na siebie większej liczby komórek, niż ma to miejsce w przypadku uszkodzenia CF, jest model uszkodzenia uwarunkowanego zawartością – PSF. Uszkodzenie PSF jest najbardziej ogólnym modelem opisującym niezamierzone interakcje pomiędzy komórkami pamięci. Może ono być rozpatrywane jako uogólnienie uszkodzenia sprzężeniowego [8]. Model ten składa się z grupy komórek będących agresorami (komórkami wiążącymi) i jednej komórki będącej ofiarą (komórką bazową). W uszkodzeniu PSF stan bazowej komórki pamięci (wartość, lub zmiana wartości) zależny jest od stanów wszystkich pozostałych komórek (komórek wiążących). Testowanie pamięci pod kątem powyższego uszkodzenia jest jednak niepraktyczne czy wręcz niemożliwe z uwagi na czas niezbędny do realizacji takiego testu. Bardziej praktyczny jest model uszkodzenia uwarunkowanego zawartością sąsiadów NPSF. W odróżnieniu od uszkodzenia PSF w uszkodzeniu NPSF przyjmuje się, iż na komórkę bazową oddziałują tylko komórki z jej najbliższego fizycznego sąsiedztwa. Opierając się na powyższym modelu zostało zaproponowanych wiele klasycznych rozwiązań, np. w [8, 10]. Jednakże w przypadku współczesnych pamięci, algorytmy te charakteryzują się zbyt długim czasem wykonania. W rezultacie opracowano wiele nowych propozycji odnoszących się do uszkodzeń NPSF, np. [5, 8, 11, 17]. Niestety, rozwiązania oparte na modelu NPSF nie zawsze mogą być wykorzystane. Źródłem problemu jest optymalizacja stosowana podczas produkcji układów cyfrowych, której wynikiem jest różnica pomiędzy logicznym (widzianym przez użytkownika) i fizycznym ułożeniem komórek pamięci względem siebie (ang. *scrambling*), [18]. Dlatego do efektywnego wykorzystania powyższych rozwiązań niezbędny staje się schemat pozwalający określić fizyczne rozmieszczenie komórek pamięci (ang. *scrambling information*), [1, 8]. Schemat ten nie zawsze jest upubliczniany przez producentów, ponadto mechanizmy redundancji (ang. *memory row and column redundancy*) sprawiają, że konfiguracja pamięci może ulec zmianie w sposób niezauważalny dla użytkownika podczas jej normalnej pracy [18]. Jednocześnie, w przypadku rozpatrywanych uszkodzeń złożonych, koniecznym warunkiem (w większości wypadków wystarczającym) wykrycia (lub aktywacji) wszystkich potencjalnych k -komórkowych uszkodzeń w wybranych k komórkach pamięci, jest wygenerowanie wszystkich 2^k możliwych binarnych kombinacji w tych komórkach. Dlatego w sytuacji, gdy nie jest znana topologia testowanej pamięci jako model uszkodzenia można przyjąć model PSF $_k$, będący uogólnieniem modelu NPSF. W modelu PSF $_k$ rozpatrywanych jest dowolne k z N komórek pamięci (N – rozmiar pamięci), wśród których jedna jest komórką bazową, a pozostałe $k - 1$ – komórkami wiążącymi. Należy

zaznaczyć, iż rezultaty otrzymane dla tak zdefiniowanego modelu, można łatwo odnieść do innych modeli, gdyż jest on najbardziej złożony w procesie wykrywania uszkodzeń pamięci [19].

Wiele różnych algorytmów zostało zaproponowanych w obszarze testowania pamięci RAM. Kluczowe wydają się podejścia oparte na testach krokowych nazywanych również maszerującymi (ang. *march tests*). Ich największymi zaletami są: złożoność liniowa w stosunku do rozmiaru pamięci, wysoki poziom pokrycia uszkodzeń i regularność struktury umożliwiająca stosunkowo prostą ich implementację wbudowaną (ang. *built-in self-test* – BIST), [8]. Jednak tradycyjne testy krokowe nie generują wszystkich możliwych wzorców niezbędnych do detekcji uszkodzeń PSFk. Dlatego zaproponowano techniki, wykorzystujące w swej podstawie testy krokowe, umożliwiające zwiększenie pokrycia uszkodzeń PSFk. Jednym z takich podejść jest testowanie wieloprzebiegowe. Technika ta polega na wielokrotnej realizacji zadanego testu przy jednoczesnej zmianie warunków początkowych (np. zawartości pamięci i/lub sekwencji adresowej).

W przypadku pojedynczego przebiegu testu krokowego warunki początkowe nie mają wpływu na finalne pokrycie uszkodzeń [8,12]. Dla dowolnej sekwencji adresowej oraz dowolnej zawartości pamięci liczba wykrywanych uszkodzeń jest zawsze taka sama i może zostać określona na podstawie przyjętych modeli uszkodzeń. W przypadku testów wieloprzebiegowych w literaturze dziedziny jasno wskazano, iż zmiana warunków początkowych w sposób znaczący wpływa na finalne pokrycie uszkodzeń testu. Dlatego jednym z kluczowych czynników wpływających na pokrycie uszkodzeń w przypadku wieloprzebiegowych testów krokowych są sekwencje adresowe kolejnych iteracji testów oraz wzajemne relacje między tymi sekwencjami. [13]. Różne sekwencje adresowe w kolejnych iteracjach testu pozwalają wygenerować nowe wzorce w komórkach pamięci co umożliwia aktywację i wykrycie dodatkowych uszkodzeń w porównaniu z iteracjami wcześniejszymi. Na przykład w przypadku dwóch przebiegów testu pamięci, musimy wybrać dwie sekwencje adresowe, które muszą się różnić pod względem zdolności wykrywania różnych konfiguracji uszkodzeń PSF. Jak zostało wykazane, różne pary sekwencji adresowych skutkują w różnym pokryciu uszkodzeń [16]. Niezmiernie ważnym problemem jest zatem wybranie odpowiednich sekwencji adresowych, uwzględniając przy tym nie tylko pokrycie uszkodzeń, ale również koszt wygenerowania tych sekwencji.

W dalszej analizie będzie rozważany model uszkodzenia PSFk i rozpatrywana pamięć RAM o pojemności $N = 2^m$ bitów, gdzie m jest dodatnią liczbą całkowitą oznaczającą liczbę linii adresowych pamięci. Jak pokazują liczne prace van de Goor'a, Nicolaidis'a i innych, w przypadku złożonych uszkodzeń pamięci, takich jak PSFk, pokrycie uszkodzeń na poziomie 100% można osiągnąć w bardzo ograniczonych przypadkach. W ogólnym przypadku 100% pokrycie złożonych uszkodzeń pamięci można osiągnąć poprzez realizację nieskończonej liczby testów lub wykorzystując test o złożoności rzędu 2^N , gdzie N to rozmiar pamięci. Dlatego we współczesnych badaniach związanych z detekcją złożonych uszkodzeń pamięci RAM dąży się do uzyskania

wysokiego pokrycia uszkodzeń przy umiarkowanej złożoności testów. Stąd badania autora zostały skupione na bardzo krótkich, dwuprzebiegowych testach krokowych. Głównymi wynikami są:

- wyznaczenie parametrów umożliwiających wygenerowanie optymalnych (w kontekście pokrycia uszkodzeń) sekwencji licznikowych jako sekwencji adresowych w dwuprzebiegowych testach krokowych,
- zaproponowanie rozwiązania umożliwiającego implementację, o niskiej złożoności sprzętowej, generatora optymalnych sekwencji adresowych,
- dokładna analityczna analiza proponowanego rozwiązania w kontekście detekcji uszkodzeń PSF.

Niski narzut sprzętowy w proponowanym rozwiązaniu osiągnięto przez implementację generatora adresów w oparciu o licznik sekwencyjny. Generator adresów posiada dwa tryby pracy. W pierwszej iteracji testu działa on jak standardowy licznik realizując działania $(+1)$ i (-1) (pierwszy tryb pracy). W przypadku drugiej iteracji sekwencja adresowa generowana jest w oparciu o licznik modyfikujący swój stan zgodnie z indeksem inkrementacji q : $(+q)$ i $(-q)$ (drugi tryb pracy). Przeprowadzona analiza pozwoliła określić optymalną wartość indeksu inkrementacji q jak również adres początkowy drugiej sekwencji adresowej umożliwiające uzyskanie wysokiego pokrycia uszkodzeń PSF k przez dwuprzebiegowy test krokowy.

4.1. Efektywność testów krokowych

Kluczową rolę w testowaniu pamięci odgrywają testy krokowe (maszerujące), [8]. Test krokowy składa się ze skończonej liczby sekwencji elementów typu March. Element typu March składa się ze skończonej liczby sekwencji operacji czytania (r) i pisania (w), z których wszystkie oddziałują na określoną komórkę przed przejściem do następnej komórki pamięci. Komórka następna określona jest zgodnie z sekwencją adresową oznaczaną przez \uparrow (nazywana często sekwencją rosnącą), lub \downarrow (nazywaną często sekwencją malejącą). Uszeregowanie adresów w sekwencji adresowej może być dowolne, przy założeniu jednak, iż w sekwencji adresowej oznaczonej symbolem \uparrow kolejność adresów jest odwrotna do sekwencji oznaczonej symbolem \downarrow [8, 15, 24]. Symbol \updownarrow wykorzystywany jest w przypadku, gdy możemy przyjąć dowolnie albo sekwencję rosnącą albo sekwencję malejącą. Możliwe działania odnoszące się do komórek pamięci dostępne w testach krokowych to: wx – zapisz do komórki pamięci wartość x i rx – odczytaj wartość z komórki pamięci (wartość spodziewana jest równa x). Cały test krokowy ograniczony jest przez parę nawiasów klamrowych $\{ \dots \}$, podczas gdy element typu March ograniczony jest parą nawiasów okrągłych (\dots) . Jako przykład rozpatrzmy test MATS+ [23]:

$$\{ \Downarrow(w0); \Uparrow(r0, w1); \Downarrow(r1, w0) \}.$$

M0 M1 M2

Test ten składa się z trzech elementów typu March: M0, M1, M2. Element M0 zeruje pamięć. Element M1, w porządku sekwencji rosnącej odpowiednio czyta wartość 0 z komórki, następnie zapisuje tam wartość 1. Element M2 testu MATS+, w porządku adresów sekwencji malejącej odczytuje wartość 1 z komórki, po czym do tej samej komórki wpisuje wartość 0. Jednymi z najbardziej znanych testów krokowych są testy MATS++ $\{ \Downarrow(w0); \Uparrow(r0, w1); \Downarrow(r1, w0, r0) \}$ i March C- $\{ \Downarrow(w0); \Uparrow(r0, w1); \Uparrow(r1, w0); \Downarrow(r0, w1); \Downarrow(r1, w0); \Downarrow(r0) \}$. W przypadku tradycyjnych testów korkowych pierwsza faza testu, ustala początkową zawartość pamięci. Dlatego testy krokowe w postaci klasycznej nie są predysponowane do stosowania w procedurach testowych uruchamianych w systemach w trakcie ich normalnej pracy (ang. *online testing*). W tym celu należy wykorzystać testy transparentne (ang. *transparent tests*).

Technika transparentnego testowania pamięci jest dobrze znanym podejściem umożliwiającym przeprowadzenie procesu testowania pamięci w sposób przezroczysty dla jej zawartości. Zaproponowana została w 1986 roku przez B. Koenemana [20]. Technikę transformacji klasycznych testów krokowych w testy transparentne zaproponował Nicolaidis [22]. Testy transparentne umożliwiają zachowanie niezmięnionej zawartości pamięci po zakończeniu testu w stosunku do zawartości z momentu rozpoczęcia testu. Dzięki temu są one szczególnie predysponowane do realizacji periodycznych testów wykonywanych w czasie normalnej pracy urządzenia (ang. *periodic field testing*). Wybrane rozwiązania oparte na testach transparentnych można znaleźć m.in. w [14, 17, 21]. Transparentne odpowiedniki testów MATS++ i March C- mają odpowiednio postać $\{ \Uparrow(ra, w\bar{a}); \Downarrow(r\bar{a}, wa, ra) \}$ i $\{ \Uparrow(ra, w\bar{a}); \Uparrow(r\bar{a}, wa); \Downarrow(ra, w\bar{a}); \Downarrow(r\bar{a}, wa); \Downarrow(ra) \}$, gdzie $a \in \{0, 1\}$ i \bar{a} jest inwersją a . Inne testy krokowe, ich złożoności i efektywność w odniesieniu do uszkodzeń prostych przedstawia tabela 4.1 [8].

TABELA 4.1. Pokrycie uszkodzeń prostych przez wybrane testy krokowe

Test	Pokrycie uszkodzeń							Złożoność
	SAF	AF	TF	CFin	CFid	CFdyn	SCF	
MATS	+							4N
MATS+	+	+						5N
MATS++	+	+	+					6N
March X	+	+	+	+				6N
March C-	+	+	+	+	+	+	+	10N
March A	+	+	+	+				15N
March Y	+	+	+	+				8N
March B	+	+	+	+				17N

„+” oznacza pełne pokrycie danego typu uszkodzeń przez test.

Należy podkreślić, iż czas realizacji testu zależy nie tylko od samego testu, ale również od technologii, w jakiej została zrealizowana pamięć. Dlatego do opisu czasu trwania testu powszechnie używana jest złożoność testu, przez którą rozumie się liczbę poleceń r/w występujących w teście [8,22,25,31]. Wartość ta jest bezpośrednio skorelowana z czasem niezbędnym do realizacji testu i jest jednocześnie niezależna od technologii. Dane, które pokazuje tabela 4.1 wskazują, iż testy krokowe posiadają wysoką efektywność w odniesieniu do uszkodzeń prostych. Jednocześnie w przypadku uszkodzeń złożonych efektywność ta jest niewystarczająca. Dla testu złożonego z jednej iteracji testu MATS++ pokrycie uszkodzeń PSFk można określić jako $FC_{MATS++}(PSFk) = (1/2^{k-1})100\%$, a w przypadku testu March C- $FC_{MarchC-}(PSFk) = (1/2^{k-2})100\%$ [8,27]. Przykładowo, $FC_{MATS++}(PSF3) = (1/2^2)100\% = 25\%$, $FC_{MATS++}(PSF5) = (1/2^4)100\% = 6,25\%$, $FC_{MarchC-}(PNPSF3) = (1/2^1)100\% = 50\%$, i $FC_{MarchC-}(PNPSF5) = (1/2^3)100\% = 12,5\%$.

4.2. Dwuprzebiegowe testy krokowe

Jak wskazano we wprowadzeniu, do uzyskania wysokiego pokrycia uszkodzeń PSFk niezbędne jest wykorzystanie optymalnych sekwencji adresowych [27, 30]. W przypadku dwuprzebiegowej sesji testowej wykorzystywane są dwie sekwencje adresowe A_j i A_k , gdzie $A_l = A_l(0)A_l(1)A_l(2) \dots A_l(N-2)A_l(N-1)$; ($A_l(i) \in \{0, 1, 2, \dots, N-1\}$, $i \in \{0, 1, 2, \dots, N-1\}$). Należy zauważyć, iż liczba możliwych różnych sekwencji adresowych A_l , $l \in \{0, 1, 2, \dots, N-1\}$ jest bardzo duża i równa $N! = 2^m!$ [27].

W celu uzyskania wysokiego pokrycia uszkodzeń PSFk dwuprzebiegowego testu krokowego niezbędne jest, aby dwie przyjęte sekwencje adresowe A_j i A_k , w jak największym stopniu różniły się między sobą [25]. W najprostszym przypadku oznacza to, iż na tych samych pozycjach i w sekwencjach adresowych A_j i A_k , powinny być różne wartości. Wymóg ten spełniony jest również w standardowych testach krokowych [8]. W kolejnych fazach testu krokowego generowane są dwie różne sekwencje adresowe: rosnąca $0, 1, 2, \dots, 2^m-2, 2^m-1$ (\Uparrow) i malejąca $2^m-1, 2^m-2, 2^m-3, \dots, 1, 0$ (\Downarrow). Jako miara zróżnicowania sekwencji adresowych może zostać wykorzystana metryka Manhattan:

$$D_{Manh}(A_j, A_k) = \sum_{i=0}^{N-1} |A_j(i) - A_k(i)|, \quad (4.1)$$

która została przeanalizowana i eksperymentalnie sprawdzona w [27].

Odległość miejska $D_{Manh}(A_j, A_k)$ pomiędzy dwoma sekwencjami adresowymi A_j i A_k , może zostać wykorzystana jako charakterystyka numeryczna umożliwiająca określenie poziomu zróżnicowania dwóch sekwencji adresowych. Przykładowo, dla dwóch sekwencji $A_j = 0, 1, 2, \dots, 2^m-1, 2^m-1$ i $A_k = 2^m-1, 2^m-2, 2^m-3, \dots, 1, 0$ powyższa odległość jest równa $D_{Manh}(A_j, A_k) = 2^{2m-1}$. Jak zostało udowodnione w [29] wartość 2^{2m-1} jest

maksymalną możliwą wartością $\max D_{Manh}(A_j, A_k)$ powyższej metryki dla dwóch różnych sekwencji adresowych A_j, A_k . Jednocześnie wartością minimalną $\min D_{Manh}(A_j, A_k)$ jest wartość 2.

Istnieje wiele rozwiązań umożliwiających generowanie sekwencji adresowych. Wszystkie opierają się na prostych algorytmach, takich jak generowanie sekwencji pseudolosowych, negacja bitów adresu, przesunięcie i/lub permutacje bitów adresu oraz wiele różnych sekwencji liczbowych opartych na kodzie Gray'a, odwrotnym kodzie Gray'a, maksymalizacji średniej odległości Hamminga i innych [27, 29, 30, 32]. Wszystkie te rozwiązania były proponowane w kontekście zapewnienia niskiej złożoności sprzętowej implementacji mechanizmów *testowania wbudowanego BIST*, jak również osiągnięcia jak najwyższego pokrycia uszkodzeń w tym PSFk.

Przykładowe pokrycie uszkodzeń PSFk przez dwuprzebiegowy test wykorzystujący pseudolosowe sekwencje adresowe przedstawia tabela 4.2. Kolumna „Iteracja pierwsza” zawiera pokrycie uszkodzeń po pierwszej iteracji testu, zaś kolumna „Iteracja druga” przedstawia sumaryczne pokrycie po wykonaniu drugiej iteracji testu. W obu przypadkach (zarówno dla testu MATS++ jak i March C-) sekwencje adresowe generowane były oparte na rejestrze przesuwającym LFSR stowarzyszonym z wielomianem pierwotnym $\varphi(x)$ [26,28,33]. W przypadku tego eksperymentu wartość miary zróżnicowania dwóch losowych sekwencji adresowych A_j i A_k może być aproksymowana przez $N^2/4$ [33].

TABELA 4.2. Pokrycie uszkodzeń PSFk (%) dwuprzebiegowego testu krokowego z pseudolosowymi sekwencjami adresowymi

Test	PSF3		PSF5	
	Iteracja pierwsza	Iteracja druga	Iteracja pierwsza	Iteracja druga
MATS++	24,90	42,74	6,31	11,87
March C-	49,87	72,74	12,46	22,15

W kolejnym eksperymencie (tabela 4.3) jako sekwencję adresową w pierwszej iteracji testu wykorzystano standardową sekwencję licznikową A_j , zaś w drugiej iteracji wykorzystaną sekwencję A_k będącą modyfikacją sekwencji A_j . Modyfikacja polegała na negacji najmniej znaczącego bitu w każdym adresie sekwencji adresowej A_j . W tym przypadku wartość zróżnicowania sekwencji adresowych jest równa $D_{Manh}(A_j, A_k) = N$, co jest wartością znacząco mniejszą w porównaniu z wynikiem poprzednim.

TABELA 4.3. Pokrycie uszkodzeń PSFk (%) dwuprzebiegowego testu krokowego z drugą sekwencją adresową wygenerowaną przez modyfikację pierwszej sekwencji (negacja bitu LSB pierwszej sekwencji)

Test	PSF3		PSF5	
	Iteracja pierwsza	Iteracja druga	Iteracja pierwsza	Iteracja druga
MATS++	25,08	25,84	6,20	11,87
March C-	50,10	51,72	12,57	13,36

Wyniki, które zawierają tabela 4.2 i tabela 4.3 wskazują na dużą zależność pokrycia uszkodzeń od wartości zróżnicowania sekwencji adresowych. W przypadku dwóch sekwencji adresowych A_j i A_k , dla których wartość zróżnicowania $D_{Manh}(A_j, A_k)$ jest duża pokrycie uszkodzeń jest znacznie większe w porównaniu z sytuacją, gdy wartość zróżnicowania sekwencji adresowych jest stosunkowo niska.

4.2.1. Dwuprzebiegowe testy krokowe z indeksem inkrementacji $q = 2$

W punkcie tym zostanie przeanalizowany przypadek, gdzie druga sekwencja adresowa testu dwuprzebiegowego zostanie wygenerowana na podstawie pierwszej sekwencji z wykorzystaniem indeksu inkrementacji o wartości $q = 2$. Jako pierwszą sekwencję adresową przyjmijmy dowolną sekwencje liczbową $A_j(i)$, gdzie $A_j(i) \in \{0, 1, 2, \dots, 2^m - 1\}$, $i \in \{0, 1, 2, \dots, 2^m - 1\}$. Wtedy druga sekwencja adresowa będzie miała postać $A_k = A_k(0)A_k(1)A_k(2) \dots A_k(2^m - 2)A_k(2^m - 1)$ zdefiniowaną zgodnie z poniższą definicją.

Definicja 4.1: Sekwencja adresowa A_k będąca wynikiem inkrementacji sekwencji adresowej $A_j = A_j(0)A_j(1)A_j(2) \dots A_j(2^m - 2)A_j(2^m - 1)$ zgodnie z indeksem $q = 2$ będzie miała następująca postać:

$$\begin{aligned}
 A_k(i) &= A_j(2i), & i \in \{0, 1, 2, \dots, 2^{m-1} - 1\}; \\
 A_k(i) &= A_j(2(i - 2^{m-1}) + 1), & i \in \{2^{m-1}, 2^{m-1} + 1, 2^{m-1} + 2, \dots, 2^m - 1\}.
 \end{aligned}
 \tag{4.2}$$

Należy zauważyć, iż Definicja 4.1 może zostać wykorzystana w odniesieniu do dowolnej sekwencji adresowej A_j , w tym standardowej sekwencji licznikowej $A_j = A_j(0)A_j(1)A_j(2) \dots A_j(2^m - 2)A_j(2^m - 1) = 0, 1, 2, \dots, 2^m - 2, 2^m - 1$. Przykładowo, dla standardowej sekwencji licznikowej dla $m = 4$ $A_j = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15$ sekwencja wynikowa A_k przyjmie wartość $0, 2, 4, 6, 8, 10, 12, 14, 1, 3, 5, 7, 9, 11, 13, 15$.

W zależności od adresu początkowego s , istnieje $N = 2^m$ wersji sekwencji A_k otrzymanych w oparciu o standardową sekwencję licznikową $A_j = A_j(0)A_j(1)A_j(2) \dots A_j(2^m - 2)A_j(2^m - 1) = 0, 1, 2, \dots, 2^m - 2, 2^m - 1$. Wszystkie sekwencje adresowe w zależności od adresu początkowego s dla $m = 3$ i indeksu inkrementacji $q = 2$, wygenerowane na podstawie standardowej sekwencji licznikowej zawiera tabela 4.4.

TABELA 4.4. Zbiór sekwencji adresowych wygenerowanych na podstawie standardowej sekwencji licznikowej dla $m = 3$ i indeksu inkrementacji $q = 2$

l	$A_j(i)$	A_k							
		$s = 0$	$s = 1$	$s = 2$	$s = 3$	$s = 4$	$s = 5$	$s = 6$	$s = 7$
0	0	0	2	4	6	1	3	5	7
1	1	2	4	6	1	3	5	7	0
2	2	4	6	1	3	5	7	0	2
3	3	6	1	3	5	7	0	2	4
4	4	1	3	5	7	0	2	4	6
5	5	3	5	7	0	2	4	6	1
6	6	5	7	0	2	4	6	1	3
7	7	7	0	2	4	6	1	3	5

Jak zostało już wskazane, w przypadku testu dwuprzebiegowego muszą zostać wybrane dwie optymalne sekwencje adresowe [27, 29, 30]. Jako miara dopasowania może zostać wykorzystana metryka miejska [27]. Wartości metryk $D_{Manh}(A_j, A_k(s))$ dla $m = 3$ i $m = 4$ zawiera tabela 4.5 i tabela 4.6.

TABELA 4.5. Wartości metryki różnicowania dla $q = 2$ i $m = 3$

s	0	1	2	3	4	5	6	7
$D_{Manh}(A_j, A_k(s))$	12	20	20	24	24	24	24	20

TABELA 4.6. Wartości metryki różnicowania dla $q = 2$ i $m = 4$

s	0	1	2	3	4	5	6	7
$D_{Manh}(A_j, A_k(s))$	12	20	20	24	24	24	24	20
s	$s = 8$	$s = 9$	$s = 10$	$s = 11$	$s = 12$	$s = 13$	$s = 14$	$s = 15$
$D_{Manh}(A_j, A_k(s))$	96	96	96	92	92	84	84	72

Dane zawarte w powyższych tabelach pozwalają stwierdzić, iż w przypadku $m = 3$ zbiór optymalnych par sekwencji adresowych A_j i $A_k(s)$ to $\{(A_j, A_k(3)), (A_j, A_k(4)), (A_j, A_k(5)), (A_j, A_k(6))\}$, zaś w przypadku $m = 4$ zbiór par sekwencji optymalnych to $\{(A_j, A_k(7)), (A_j, A_k(8)), (A_j, A_k(9)), (A_j, A_k(10))\}$.

Z uwagi na regularną strukturę sekwencji adresowej $A_k(s)$ nie jest trudno pokazać, iż wartość powyższej metryki zróżnicowania, dla dowolnego rozmiaru pamięci $N = 2^m$ oraz dowolnego adresu początkowego s , można wyznaczyć na podstawie (4.3).

$$D_{Manh}(A_j, A_k(s)) = \begin{cases} \frac{s(2^m - s + 2)}{2} + 2^{2m-2} - 2^{m-1}, & \text{dla parzystych } s; \\ \frac{s(2^m - s) + 1}{2} + 2^{2m-2}, & \text{dla nieparzystych } s. \end{cases} \quad (4.3)$$

Optymalna wartość adresu początkowego s w kontekście otrzymania największej wartości metryki zróżnicowania $D_{Manh}(A_j, A_k(s))$ dla parzystego s można otrzymać na podstawie (4.4).

$$\frac{dD_{Manh}(A_j, A_k(s))}{ds} = 2^{m-1} - s = 0. \quad (4.4)$$

Najbliższe parzyste wartości będące rozwiązaniem powyższego równania to $s = 2^{m-1}$ i $s = 2^{m-1} + 2$. W przypadku $m = 3$ i $m = 4$ są to wartości odpowiednio 4,6 oraz 8,10 (porównaj tabela 4.5 i tabela 4.6).

Optymalny nieparzysty adres początkowy s może zostać wyznaczony na podstawie (4.5).

$$\frac{dD_{Manh}(A_j, A_k(s))}{ds} = 2^{m-1} - s = 0. \quad (4.5)$$

Najbliższe nieparzyste wartości będące rozwiązaniem powyższego równania to $s = 2^{m-1} - 1$ oraz $s = 2^{m-1} + 1$. W przypadku $m = 3$ i $m = 4$ są to wartości odpowiednio 3,5 oraz 7,9 (porównaj tabela 4.5 i 4.6).

4.2.2. Dwuprzebiegowe testy krokowe z indeksem inkrementacji $q = 3$

Dla pamięci o rozmiarze $N = 2^m$ bitów, gdzie m dowolna całkowita liczba dodatnia, zbiór adresów dostępnych komórek pamięci to $A_i(i) \in \{0, 1, 2, \dots, 2^m - 1\}$, $i \in \{0, 1, 2, \dots, 2^m - 1\}$. Przed zdefiniowaniem sekwencji adresowej będącej rezultatem inkrementacji sekwencji bazowej o indeks $q = 3$, udowodnione zostanie stwierdzenie 4.1.

Stwierdzenie 4.1: Relacja $(2^m - 1) \bmod 3 \neq 2$ jest prawdziwa dla dowolnej całkowitej dodanej wartości m .

Dowód:

Założmy, że $(2^m - 1) \bmod 3 = 2$, wówczas $(2^m - 1) = 3p + 2$, gdzie p jest dodatnią liczbą całkowitą. Po przekształceniu otrzymamy $3p = 2^m - 3$. Z ostatniego równania wynika, iż 2^m jest podzielne przez 3 co nie jest prawdą, dlatego $(2^m - 1) \bmod 3 \neq 2$. □

W następstwie powyższego stwierdzenia można przedstawić Własność 4.1 i Własność 4.2.

Własność 4.1: $(2^m - 1) \bmod 3 \in \{0, 1\}$.

Uwzględniając Własność 4.1 oraz równanie 4.6

$$(a \otimes b) \bmod d = ((a \bmod d) \otimes (b \bmod d)) \bmod d, \quad (4.6)$$

dla liczb całkowitych a, b i d , gdzie $\otimes \in \{+, \times\}$, prawdziwa jest własność 4.2.

Własność 4.2: $((2^m - 1) + 2) \bmod 3 = (2^m + 1) \bmod 3 \in \{0, 2\}$.

Stwierdzenie 4.2: Dla parzystych m $(2^m - 1) \bmod 3 = 0$, zaś dla nieparzystych m $(2^m - 1) \bmod 3 = 1$

Dowód:

Założmy, iż $m = 2n$ jest parzystą liczbą całkowitą, wówczas $(2^m - 1) = (2^{2n} - 1) = (2^n + 1)(2^n - 1)$. Biorąc pod uwagę Własność 4.1 otrzymujemy $(2^n - 1) \bmod 3 = 0$ lub 1 i jednocześnie $(2^n + 1) \bmod 3$ odpowiednio równe jest 2 lub 0. Biorąc po uwagę założenie początkowe oraz równanie 4.6 otrzymujemy $(2^m - 1) \bmod 3 = ((2^n + 1) \times (2^n - 1)) \bmod 3 = ((2^n + 1) \bmod 3) \times ((2^n - 1) \bmod 3) \bmod 3$.

Rozważmy nieparzystą wartość m . Jest łatwo zauważyć, iż $(2 \times (2^m - 1)) \bmod 3 = ((2^{m+1} - 1) - 1) \bmod 3 = 2$. Można to stwierdzić biorąc pod uwagę, że $(m + 1)$ jest parzyste, a $(-1) \bmod 3 = 2$. Dlatego w rozpatrywanym przypadku $(2^m - 1) \bmod 3$ jest równe 1. □

Biorąc pod uwagę Stwierdzenie 4.2, zdefiniowana zostanie postać sekwencji adresowej będącej wynikiem inkrementacji sekwencji bazowej o indeks $q = 3$.

Definicja 4.2: Sekwencja adresowa A_k będąca wynikiem inkrementacji sekwencji adresowej $A_j = A_j(0)A_j(1)A_j(2)\dots A_j(2^m - 2)A_j(2^m - 1)$ zgodnie z indeksem $q = 3$ dla dowolnej sekwencji A_j i $A_j = 0, 1, 2, \dots, 2^m - 2, 2^m - 1$ będzie miała następującą postać:

$$A_k(i) = A_j(3i \bmod (2^m - 1)) = 0, 3, 6, \dots, 2^m - 4, 2^m - 1, 2, 5, 8, \dots, 2^m - 5, 2^m - 2, 1, 4, 7, \dots, 2^m - 6, 2^m - 3; \quad \text{dla parzystych } m, \quad (4.7)$$

$$A_k(i) = A_j(3i \bmod (2^m - 1)) = 0, 3, 6, \dots, 2^m - 5, 2^m - 2, 1, 4, 7, \dots, \\ 2^m - 4, 2^m - 1, 2, 5, 8, \dots, 2^m - 6, 2^m - 3; \quad \text{dla nieparzystych } m. \quad (4.8)$$

Wartości otrzymanych metryk $D_{Manh}(A_j, A_k(s))$ przedstawiają tabela 4.7 i tabela 4.8. Krótka analiza wyników zawartych w tych tabelach pozwala stwierdzić, że wartość metryki zróżnicowania silnie zależy od adresu początkowego s .

TABELA 4.7. Wartości metryki zróżnicowania dla $q = 3$ i $m = 3$

s	0	1	2	3	4	5	6	7
$D_{Manh}(A_j, A_k(s))$	16	20	16	24	24	20	24	24

TABELA 4.8. Wartości metryki zróżnicowania dla $q = 3$ i $m = 4$

s	$s = 0$	$s = 1$	$s = 2$	$s = 3$	$s = 4$	$s = 5$	$s = 6$	$s = 7$
$D_{Manh}(A_j, A_k(s))$	72	68	72	88	80	84	96	88
s	$s = 8$	$s = 9$	$s = 10$	$s = 11$	$s = 12$	$s = 13$	$s = 14$	$s = 15$
$D_{Manh}(A_j, A_k(s))$	88	100	88	88	96	84	80	88

Zdefiniowana zostanie teraz metryka $D_{Manh}(A_j, A_k(s))$ jako funkcja rozmiaru pamięci i adresu początkowego sekwencji adresowej będącej wynikiem inkrementacji sekwencji bazowej zgodnie z indeksem $q = 3$.

W przypadku parzystych wartości m zastosujemy następujące podstawienie: $a = (N - 1)/3$. Wówczas kolejne adresy sekwencji A_j i $A_k(s = 0)$ można zapisać w postaci:

$$A_j = 0, 1, \dots, a, a+1, a+2, (3a+1)/2, (3a+1)/2+1, (3a+1)/2+2, \dots, 2a, 2a+1, 2a+2, \dots, 3a. \quad (4.9)$$

$$A_k(s = 0) = 0, 3, \dots, 3a, 2, 5, \dots, (3a+1)/2, (3a+1)/2+3, (3a+1)/2+6, \dots, 3a-1, 1, 4, \dots, 3a-2$$

Następnie, dla parzystej wartości m i $N = 2^m$ wartość miary zróżnicowania można wyznaczyć jako:

$$D_{Manh}(A_j, A_k(s = 0)) = \sum_{i=0}^{N-1} |A_j(i) - A_k(i)| = 2 + 4 + \dots + 2a + 2 + 4 + \dots + a - 1 + 2 + 4 + \dots \\ \dots + a - 1 + 2 + 4 + \dots + 2a = \frac{5a^2 + 4a - 1}{2} = \frac{5N^2 + 2N - 16}{18}. \quad (4.10)$$

Podobnie w przypadku nieparzystej wartości m można pokazać, iż wartość miary zróżnicowania sekwencji adresowych może być wyznaczona poprzez:

$$D_{Manh}(A_j, A_k(s = 0)) = \sum_{i=0}^{N-1} |A_j(i) - A_k(i)| = \frac{5N^2 - 2N - 16}{18}. \quad (4.11)$$

Bardziej złożona sytuacja jest w przypadku $s \neq 0$. W przypadku parzystych wartości m wartość miary zróżnicowania można wyznaczyć z:

$$D_{Manh}(A_j, A_k(s)) = \begin{cases} \frac{5 \times 2^{2m} + 2^{m+1} + 3 \times 2^{m+1} \times s - 16 + 12 \times s - 6 \times s^2}{18}, & s \in \{0, 6, 12, \dots, 2^m - 4\}; \\ \frac{5 \times 2^{2m} + 2^{m+1} + 3 \times 2^{m+1} \times s + 2 + 12 \times s - 6 \times s^2}{18}, & s \in \{3, 9, 15, \dots, 2^m - 1\}; \\ \frac{5 \times 2^{2m} - 5 \times 2^{m+1} + 3 \times 2^{m+1} \times s - 16 + 12 \times s - 6 \times s^2}{18}, & s \in \{2, 4, 8, \dots, 2^m - 2\}; \\ \frac{5 \times 2^{2m} - 5 \times 2^{m+1} + 3 \times 2^{m+1} \times s + 2 + 12 \times s - 6 \times s^2}{18}, & s \in \{1, 5, 7, \dots, 2^m - 3\}. \end{cases} \quad (4.12)$$

Optymalne adresy początkowe s są wartościami, które maksymalizują wartość miary zróżnicowania sekwencji adresowych $D_{Manh}(A_j, A_k(s))$ i mogą być otrzymane na podstawie wyrażenia (4.13), które jest identyczne dla wszystkich czterech powyższych funkcji.

$$\frac{dD_{Manh}(A_j, A_k(s))}{ds} = \frac{3 \times 2^{m+1} + 12 - 12 \times s}{18} = 0, \quad s \in \{0, 1, 2, \dots, 2^m - 1\}. \quad (4.13)$$

Z wyrażenia (4.13) otrzymujemy, iż w zależności od wartości s dla wszystkich funkcji (4.12) maksymalna wartość zostanie otrzymana dla wartości s najbliższej do $s = 2^{m-1} + 1$. Dlatego maksymalną wartość dla pierwszej funkcji otrzymamy na podstawie $s = 2^{m-1} - 2$ ponieważ w zbiorze wartości $s \in \{0, 6, 12, \dots, 2^m - 4\}$ (4.12) jest to najbliższa wartość do $s = 2^{m-1} + 1$. Odpowiednio w przypadku drugiej funkcji optymalny adres startowy będzie równy $s = 2^{m-1} + 1$, w przypadku trzeciej funkcji $s = 2^{m-1}$ i w przypadku czwartej funkcji $s = 2^{m-1} - 1$. Po podstawieniu optymalnych wartości s do (4.12) łatwo jest pokazać, iż maksymalną wartość miary zróżnicowania sekwencji adresowych $D_{Manh}(A_j, A_k(s))$ otrzymujemy dla adresu startowego $s = 2^{m-1} + 1$. Przykładowo, dla $m = 4$ optymalna wartość $D_{Manh}(A_j, A_k(9))$ jest równa 100 dla adresu początkowego $s = 24 - 1 + 1 = 9$, co jest zgodne z wynikami, które przedstawia tabela 4.8.

W przypadku nieparzystej wartości m otrzymujemy:

$$D_{Manh}(A_j, A_k(s)) = \begin{cases} \frac{5 \times 2^{2m} - 2^{m+1} + 3 \times 2^{m+1} \times s - 16 + 12 \times s - 6 \times s^2}{18}, & s \in \{0, 4, 6, 10, \dots, 2^m - 2\}; \\ \frac{5 \times 2^{2m} - 2^{m+1} + 3 \times 2^{m+1} \times s + 2 + 12 \times s - 6 \times s^2}{18}, & s \in \{1, 3, 7, 9, \dots, 2^m - 1\}; \\ \frac{5 \times 2^{2m} - 7 \times 2^{m+1} + 3 \times 2^{m+1} \times s - 16 + 12 \times s - 6 \times s^2}{18}, & s \in \{2, 8, 14, \dots, 2^m - 6\}; \\ \frac{5 \times 2^{2m} - 7 \times 2^{m+1} + 3 \times 2^{m+1} \times s + 2 + 12 \times s - 6 \times s^2}{18}, & s \in \{5, 11, 17, \dots, 2^m - 3\}; \end{cases} \quad (4.14)$$

Tak jak w poprzednim przypadku, optymalną wartością adresu początkowego s dla wszystkich czterech funkcji (4.14) jest adres najbliższy $s = 2^{m-1} + 1$. W przypadku pierwszej funkcji są to adresy $s = 2^{m-1} + 2$ i $s = 2^{m-1}$, ponieważ są one najbliższe wartości $s = 2^{m-1} + 1$ spośród $0, 4, 6, \dots, 2^m - 2$. W przypadku drugiej funkcji maksymalną wartość otrzymujemy dla $s = 2^{m-1} - 1$ i $s = 2^{m-1} + 3$, w przypadku trzeciej dla $s = 2^{m-1} - 2$ i $s = 2^{m-1} + 4$ i w przypadku czwartej dla $s = 2^{m-1} + 1$. Po podstawieniu optymalnych wartości s do (4.14) jest łatwo pokazać, iż maksymalną wartość zróżnicowania sekwencji adresowych otrzymamy dla adresów startowych $s = 2^{m-1} + 2$, $s = 2^{m-1} - 1$, $s = 2^{m-1} - 2$ i $s = 2^{m-1} + 3$. Dla $m = 3$ $D_{Manh}(A_j, A_k(6)) = D_{Manh}(A_j, A_k(3)) = D_{Manh}(A_j, A_k(4)) = D_{Manh}(A_j, A_k(7)) = 24$ (patrz tabela 4.7).

4.3. Wyniki eksperymentalne

W celu potwierdzenia otrzymanych wyników teoretycznych przeprowadzono odpowiednie badania symulacyjne. Jako pierwsze zostały wykonane badania odnośnie wskaźnika inkrementacji $q = 2$. Symulacje przeprowadzono dla pamięci o rozmiarze $N = 16$ bitów w celu porównania z danymi przedstawionymi w tabeli 4.6. Doświadczenia wykonano dla testów złożonych z dwóch iteracji opartych odpowiednio o test MATS++ i March C-. W obu przypadkach przeprowadzono symulacje dla wszystkich możliwych adresów startowych sekwencji adresowej drugiej iteracji testu. W ramach przeprowadzonych eksperymentów wyznaczono średnie pokrycie uszkodzeń PSFk. Otrzymane wartości odpowiednio dla $k = 3$ i $k = 4$ zawierają odpowiednio tabela 4.9 i tabela 4.10.

TABELA 4.9. Pokrycie uszkodzeń PSFk dla dwóch iteracji testu MATS++ i indeksu inkrementacji $q = 2$

s	$s = 0$	$s = 1$	$s = 2$	$s = 3$	$s = 4$	$s = 5$	$s = 6$	$s = 7$
FC_{MATS++}(PSF3)	34,87	37,37	38,37	40,62	41,50	42,83	43,37	44,03
FC_{MATS++}(PSF5)	10,11	10,84	11,05	11,63	11,72	11,93	11,99	12,15
s	$s = 8$	$s = 9$	$s = 10$	$s = 11$	$s = 12$	$s = 13$	$s = 14$	$s = 15$
FC_{MATS++}(PSF3)	44,20	44,18	44,00	43,36	42,77	41,52	40,58	38,66
FC_{MATS++}(PSF5)	12,13	12,14	12,05	12,03	11,95	11,75	11,53	11,12

TABELA 4.10. Pokrycie uszkodzeń PSFk dla dwóch iteracji testu March C– i indeksu inkrementacji $q = 2$

s	$s = 0$	$s = 1$	$s = 2$	$s = 3$	$s = 4$	$s = 5$	$s = 6$	$s = 7$
FC_{MarchC–}(PSF3)	66,46	69,77	69,29	71,94	71,57	73,24	73,02	73,98
FC_{MarchC–}(PSF5)	20,00	21,51	21,17	22,41	22,14	22,91	22,71	23,15
s	$s = 8$	$s = 9$	$s = 10$	$s = 11$	$s = 12$	$s = 13$	$s = 14$	$s = 15$
FC_{MarchC–}(PSF3)	73,80	73,83	73,82	73,12	73,17	71,55	71,81	69,40
FC_{MarchC–}(PSF5)	22,99	23,05	23,01	22,70	22,83	22,09	22,27	21,14

Na podstawie otrzymanych wyników można stwierdzić, iż miara zróżnicowania sekwencji adresowych $D_{Manh}(A_j, A_k(s))$ pozwala określić optymalne sekwencje adresowe dla dwuprzebiegowego testu krokowego. W przypadku indeksu inkrementacji $q = 2$ w pierwszej iteracji jako sekwencję adresową wykorzystujemy standardową sekwencję licznikową zaś w drugiej iteracji sekwencję adresową rozpoczynającą się od adresu startowego $s \in \{2^{m-1}-1, 2^{m-1}, 2^{m-1}+1, 2^{m-1}+2\}$.

Celem potwierdzenia rozważań analitycznych dla indeksu inkrementacji $q = 3$, przeprowadzono te same badania co w przypadku $q = 2$, przy czym sekwencję adresową dla drugiej iteracji testu generowano zgodnie z Definicją 2.2. Otrzymane wartości odpowiednio dla $k = 3$ i $k = 4$ zawiera odpowiednio tabela 4.11 i tabela 4.12.

TABELA 4.11. Pokrycie uszkodzeń PSFk dla dwóch iteracji testu March C– i indeksu inkrementacji $q = 3$

s	$s = 0$	$s = 1$	$s = 2$	$s = 3$	$s = 4$	$s = 5$	$s = 6$	$s = 7$
FC_{MATS++}(PPSF3)	39,61	38,49	39,59	42,93	41,52	42,15	44,93	43,20
FC_{MATS++}(PPSF5)	11,21	10,98	11,18	11,90	11,68	11,71	12,17	11,88
s	$s = 8$	$s = 9$	$s = 10$	$s = 11$	$s = 12$	$s = 13$	$s = 14$	$s = 15$
FC_{MATS++}(PPSF3)	43,32	45,64	43,37	43,13	44,92	42,20	41,47	42,93
FC_{MATS++}(PPSF5)	11,94	12,25	11,86	11,91	12,24	11,67	11,63	11,92

Porównanie wyników otrzymanych dla indeksów inkrementacji $q = 2$ i $q = 3$ nie wskazują jednoznacznie, czy zwiększenie indeksu inkrementacji wpływa na zwiększenie pokrycia uszkodzeń. Porównując wyniki z tabel 4.9 i 4.10 oraz 4.11 i 4.12 można zauważyć, iż w niektórych wypadkach pokrycie uszkodzeń nieznacznie wzrosło zaś w innych wręcz spadło. Dlatego przeprowadzono dalsze badania symulacyjne dla indeksów inkrementacji $q = 4$ i $q = 5$. Otrzymane wyniki zawierają table 4.13–4.16.

TABELA 4.12. Pokrycie uszkodzeń PSFk dla dwóch iteracji testu March C- i indeksu inkrementacji $q = 3$

s	s = 0	s = 1	s = 2	s = 3	s = 4	s = 5	s = 6	s = 7
$FC_{MarchC-}(PSF3)$	71,49	71,20	71,43	72,33	72,48	72,58	72,80	73,13
$FC_{MarchC-}(PSF5)$	21,96	21,80	21,99	22,33	22,35	22,44	22,58	22,61
s	s = 8	s = 9	s = 10	s = 11	s = 12	s = 13	s = 14	s = 15
$FC_{MarchC-}(PSF3)$	73,16	73,04	73,11	73,09	72,84	72,62	72,40	72,33
$FC_{MarchC-}(PSF5)$	22,66	22,65	22,63	22,70	22,57	22,33	22,37	22,35

TABELA 4.13. Pokrycie uszkodzeń PSFk dla dwóch iteracji testu MATS++ i indeksu inkrementacji $q = 4$

s	s = 0	s = 1	s = 2	s = 3	s = 4	s = 5	s = 6	s = 7
$FC_{MATS++}(PSF3)$	37,32	40,51	41,34	40,16	40,76	43,44	43,70	42,24
$FC_{MATS++}(PSF5)$	10,71	11,42	11,49	11,51	11,50	12,03	11,98	11,84
S	s = 8	s = 9	s = 10	s = 11	s = 12	s = 13	s = 14	s = 15
$FC_{MATS++}(PSF3)$	42,77	44,67	44,57	42,75	42,56	43,96	43,48	41,10
$FC_{MATS++}(PSF5)$	11,84	12,16	12,13	11,87	11,82	12,00	12,06	11,50

TABELA 4.14. Pokrycie uszkodzeń PSFk dla dwóch iteracji testu March C- i indeksu inkrementacji $q = 4$

S	s = 0	s = 1	s = 2	s = 3	s = 4	s = 5	s = 6	s = 7
$FC_{MarchC-}(PSF3)$	69,42	72,01	73,07	72,03	71,28	73,35	73,59	72,73
$FC_{MarchC-}(PSF5)$	21,02	22,25	22,50	22,51	21,86	22,61	22,79	22,46
S	s = 8	s = 9	s = 10	s = 11	s = 12	s = 13	s = 14	s = 15
$FC_{MarchC-}(PSF3)$	72,55	73,80	73,79	72,52	72,71	73,62	73,23	71,42
$FC_{MarchC-}(PSF5)$	22,38	22,97	22,89	22,35	22,44	22,90	22,74	21,88

TABELA 4.15. Pokrycie uszkodzeń PSFk dla dwóch iteracji testu MATS++- i indeksu inkrementacji $q = 5$

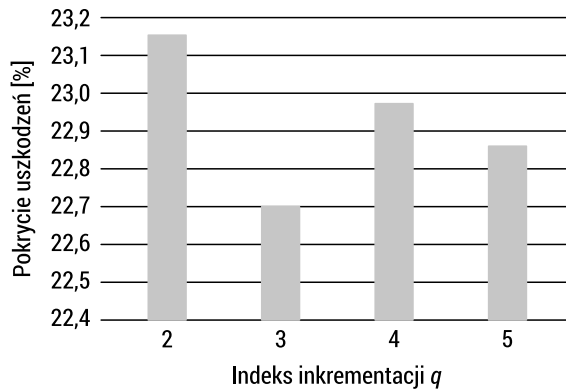
S	s = 0	s = 1	s = 2	s = 3	s = 4	s = 5	s = 6	s = 7
$FC_{MATS++}(PSF3)$	42,90	41,09	40,46	41,10	42,83	45,82	43,57	42,83
$FC_{MATS++}(PSF5)$	11,69	11,64	11,49	11,51	11,72	12,31	11,99	11,89
S	s = 8	s = 9	s = 10	s = 11	s = 12	s = 13	s = 14	s = 15
$FC_{MATS++}(PSF3)$	43,27	44,57	47,00	44,62	43,27	43,13	43,97	45,86
$FC_{MATS++}(PSF5)$	11,84	11,98	12,37	11,94	11,93	11,88	11,96	12,20

TABELA 4.16. Pokrycie uszkodzeń PSFk dla dwóch iteracji testu March C- i indeksu inkrementacji $q = 5$

S	s = 0	s = 1	s = 2	s = 3	s = 4	s = 5	s = 6	s = 7
$FC_{MarchC-}(PSF3)$	72,43	72,99	72,98	72,81	72,37	71,48	72,56	73,22
$FC_{MarchC-}(PSF5)$	22,37	22,74	22,79	22,66	22,41	22,09	22,50	22,86
s	s = 8	s = 9	s = 10	s = 11	s = 12	s = 13	s = 14	s = 15
$FC_{MarchC-}(PSF3)$	73,19	72,56	71,34	72,38	73,10	73,14	72,51	71,63
$FC_{MarchC-}(PSF5)$	22,65	22,51	21,92	22,43	22,85	22,79	22,48	21,89

Otrzymane wyniki jednoznacznie wskazują, iż zwiększanie indeksu inkrementacji nie wpływa na poziom pokrycia uszkodzeń PSFk. Na rysunku 4.1 przedstawiono maksymalne wartości pokrycia uszkodzeń otrzymane w dwóch iteracjach testu March C- dla różnych q . Z tego rysunku wynika, iż największe pokrycie uszkodzeń udało się uzyskać w przypadku $q = 2$. Jednocześnie należy podkreślić, iż we wszystkich wypadkach otrzymane wyniki są wyższe niż w przypadku wykorzystania sekwencji pseudolosowych (porównaj rysunek 4.1, tabela 4.2).

Analiza danych z tabel 4.2, 4.9 i 4.10 pozwala stwierdzić, iż w przypadku uszkodzenia PSF3 prezentowane podejście pozwala uzyskać pokrycie uszkodzeń na poziomie 44,20% i 73,80% odpowiednio przez testy MATS++ i March C-. Jednocześnie należy zauważyć, iż wynik pokrycia powyższych uszkodzeń przy adresacji pseudolosowej to odpowiednio 42,74% i 72,74%. Ważne jest aby zaznaczyć, iż w obu przypadkach złożoność testu jest identyczna.



RYSUNEK 4.1. Maksymalne pokrycie uszkodzeń dwuprzebiegowego testu March C- przy różnych wartościach q

Podsumowując otrzymane wyniki można stwierdzić, iż indeks inkrementacji $q = 2$ wydaje się być najbardziej optymalnym dla omawianej techniki testowania opartej na dwuprzebiegowych testach krokowych. Wskazuje na to wysokie pokrycie uszkodzeń zweryfikowane badaniami symulacyjnymi. Kolejnym argumentem przemawiającym na korzyść *indeksu inkrementacji* $q = 2$, jest możliwość stosunkowo łatwej implementacji sprzętowej takiego generatora adresów. Standardowo przy $q > 1$ implementacja sprzętowa generatora adresów może składać się z binarnego sumatora o rozmiarze $r = \log_2 q$ bitów oraz $(m - r)$ bitowego standardowego licznika $q = 1$. Wraz ze wzrostem q , z uwagi na większą złożoność sumatora, rośnie również narzut sprzętowy związany z implementacją generatora adresów. W przypadku $q = 2$ implementację całego generatora adresów można oprzeć wyłącznie o standardowy licznik. Przesunięcie najstarszego bitu (MSB) w standardowej sekwencji adresowej ($q = 1$) na pozycję najmłodszego bitu (LSB) pozwala osiągnąć sekwencję adresową zgodną z $q = 2$. Mechanizm ten ilustruje tabela 4.17.

TABELA 4.17. Generowanie sekwencji adresowej dla $q = 2$ na podstawie $q = 1$

Sekwencje adresowe					
$q = 1$			$q = 2$		
Q_2	Q_1	Q_0	Q_1	Q_0	Q_2
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	1	1	0
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	0	1	1
1	1	0	1	0	1
1	1	1	1	1	1

W takiej sytuacji implementacja sprzętowa generatora adresów testu dwuprzebiegowego ograniczy się do standardowego licznika generującego sekwencję adresową dla $q = 1$ oraz dodatkowo na każdej linii adresowej jednego dwuwejściowego multiplexera, co jest znacznie mniejszym narzutem sprzętowym niż przy rozwiązaniu opartym na sumatorze.

Podsumowanie

W pracy przeanalizowano dwuprzebiegowe testy krokowe pamięci RAM z inkrementacją sekwencji adresowej. Jako docelowy model uszkodzenia, na podstawie którego dokonano oceny proponowanej techniki testowania, wybrano PSFk. Podczas pierwszej iteracji testu jako bazową sekwencję adresową wykorzystano standardową sekwencję licznikową, a w przypadku drugiej iteracji były analizowane zróżnicowane sekwencje adresowe otrzymywane na podstawie inkrementacji o indeks q sekwencji bazowej oraz przez ustalenie optymalnego adresu startowego. Największe pokrycie uszkodzeń uzyskano w przypadku indeksu inkrementacji $q = 2$. W tym przypadku pokrycie uszkodzeń jest większe lub porównywalne niż dla innych wartości q . Dodatkową zaletą przemawiającą za indeksem $q = 2$ jest stosunkowo mały narzut związany z implementacją sprzętową generatora adresów. Jak zostało również dowiedzione analitycznie i sprawdzone eksperymentalnie, ważnym czynnikiem wpływającym na sumaryczne pokrycie uszkodzeń testu dwuprzebiegowego jest adres początkowy sekwencji adresowej drugiej iteracji testu. W analizowanych przykładach, różnica w pokryciu uszkodzeń przy optymalnym i nieoptymalnym doborze adresu startowego wynosiła ponad 26%.

Bibliografia

- [1] Adams R. D., *High performance memory testing: design principles, fault modeling and self-test*. luwer Academic Publishers, USA 2003
- [2] Di Carlo S., and Prinetto P., *Models in Memory Testing*. Springer Netherlands, Dordrecht, 2010, s. 157–185
- [3] Hamdioui S., Taouil M., and Haron N. Z., Testing open defects in memristor-based memories. *IEEE Transactions on Computers* 64, 1 (2015), 247–259
- [4] Hayrapetyan D., Manukvan A., and Tshagharyan G., *Implementation of memory static, coupling and dynamic fault models at the register transfer level*. [W:] 2018 IEEE East-West Design Test Symposium (EWDTS) (9 2018), 1–4
- [5] Sfikas Y., and Tsiatouhas Y., *Testing neighbouring cell leakage and transition induced faults in DRAMs*. “*Transactions on Computers*” 2016, 65, 7, 2339–2345
- [6] Thatte S., and Abraham J., *Testing of semiconductor random access memories*. Proceedings of the Annual International Conference on Fault-Tolerant Computing 1977, 81–87

- [7] Tshagharyan G., Harutyunyan G., Zorian Y., Gebregiorgis A., Golanbari M. S., Bishnoi R., and Tahoori M. B., *Modeling and testing of aging faults in finfet memories for automotive applications*. [W:] 2018 IEEE International Test Conference (ITC) 2018, 10, 1–10
- [8] Van de Goor A. J., *Testing Semiconductor Memories: Theory and Practice*. John Wiley & Sons, Chichester, England 1991
- [9] Yarmolik V. N., Klimets Y., and Demidenko S., *March PS(23N) test for DRAM pattern-sensitive faults*. [W:] Proceedings of the 7th Asian Test Symposium 1998, AT'S'98, IEEE Computer Society, 354–357
- [10] Hayes J. P., *Detection of Pattern-Sensitive Faults in Random-Access Memories*. “IEEE Transactions on Computers” 1975, 24.2, 150–157
- [11] Cockburn B. F., *Deterministic tests for detecting scrambled pattern-sensitive faults in RAMs*. Proceedings of the IEEE International Workshop on Memory Technology, Design and Testing. MTDT'95. Washington, DC, USA: IEEE Computer Society 1995, 117–122
- [12] Nicolaidis M., *Theory of Transparent BIST for RAMs*. “IEEE Transaction of Computers” 1996, 45, 10
- [13] Yarmolik S.V., *Address Sequences and Backgrounds with different Hamming Distance for Multiple run March tests*. “International Journal of Applied Mathematics and Computer Science” 2008, 1 8, 329–339
- [14] B. Cockburn F. i Sat Y. F. N., *A transparent built-in self-test scheme for detecting single V-coupling faults in RAMs*. Proceedings of the IEEE International Workshop on Memory Technology, Design, and Test 1994, 119–124. doi: 10.1109/MTDT.1994.397187
- [15] Yarmolik V. N. and Yarmolik S.V., *Address sequences for multiple run march tests*. “Autom. Control Comput. Sci.” 2006, 5, 59–68
- [16] Yarmolik S. V. and Yarmolik V. N., *Memory address generation for multiple run march tests with different average Hamming distance*. Proc. IEEE East-West Design and Test Workshop, Sochi, Russia 2006, 212–216
- [17] Yarmolik V.N., Klimets Y. i Demidenko S., *March PS(23N) Test for DRAM Pattern-Sensitive Faults Proceedings of the 7th Asian Test Symposium*. AT'S'98. “IEEE Computer Society” 1998, 354–357
- [18] Van de Goor A. J. i Schanstra I., *Address and data scrambling: causes and impact on memory tests*, Proceedings of the First IEEE International Workshop on Electronic Design, Test and Applications. 2002, 128–136, doi: 10.1109/DELTA.2002.994601
- [19] Van de Goor A. J. i in., *March LR: a test for realistic linked faults*, Proceedings of the 14th VLSI Test Symposium. Kw. 1996, 272–280. doi: 10.1109/VTEST.1996.510868
- [20] Koeneman B., *oral presentation*, Design For Testability Workshop 1986
- [21] Nicolaidis M., *Theory of Transparent BIST for RAM*, IEEE Transactions on Computers 1996, 45.10, 1141–1156. ISSN: 0018-9340
- [22] Nicolaidis M., *Transparent BIST for RAMs*. Proceedings of the IEEE International Test Conference, Discover the New World of Test and Design. ITS'92. Baltimore, Maryland, USA: IEEE Computer Society 1992, 598–607
- [23] Hamdioui S., *Testing static random access memories: defects, fault models and test patterns*, Frontiers in Electronic Testing, Kluwer Academic 2004
- [24] Niggemeyer D., Redeker M. i Otterstedt J., *Integration of Non-classical Faults in Standard March Tests*. Proceedings of the 1998 IEEE International Workshop on Memory Technology, Design and Testing, 91, IEEE Computer Society, San Jose 1998
- [25] Cheng K., Tsai M. i Wu C., *Efficient neighborhood pattern-sensitive fault test algorithms for semiconductor memories*, Proc. 19th IEEE VLSI Test Symp. (VTS '01), IEEE Computer Society, Washington, DC, 2001, 225–230

- [26] Hellebrand S., Wunderlich H.-J., Ivaniuk A. A., Klimets Y. V. i Yarmolik V. N., *Efficient online and online testing of embedded DRAMs*, "IEEE Trans. Comput." 2002, 51, 801–809
- [27] Yarmolik S., *Address sequences and backgrounds with different hamming distances for multiple run march tests* "Int. J. Appl. Math. Comput. Sci." 2008, 18, 329–339
- [28] Hellebrand S., Wunderlich H.-J. i Yarmolik V. N., *Symmetric transparent BIST for RAMs*, Proc. Conf. Design, Automation and Test in Europe ACM, New York 1999, 702–707
- [29] Yarmolik V. and Yarmolik S., *Address sequences for multiple run march tests*. "Autom. Control Comput. Sci." 2006, 5, 59–68
- [30] Yarmolik S. V. and Yarmolik V. N., *Memory address generation for multiple run march tests with different average Hamming distance*, Proc. IEEE East-West Design and Test Workshop, Sochi, Russia 2006, 212–216
- [31] Wunderlich H.-J., *Multiple distributions for biased random test patterns*, Proc. 1988 Int. Conf. Test: New Frontiers in Testing (ITC'88), IEEE Computer Society, Washington, DC, 1988, 236–244
- [32] Sokol B. i Yarmolik S. V., *Address sequences for march tests to detect pattern sensitive faults*, Proc. Third IEEE Int. Workshop Electronic Design, Test and Applications (DELTA '06), IEEE Computer Society, Kuala Lumpur, 2006, 354–360
- [33] Karpovsky M. G. i Yarmolik V. N., *Transparent memory testing for pattern-sensitive faults*, Proc. IEEE Int. Test Conf. TEST: The Next 25 Years, IEEE Computer Society, Washington, DC, 1994, 860–869

Two runs ram march testing with address decimation

Abstract: Conventional march memory tests have high fault coverage, especially for simple faults like stack-at (SAF) , transitions (TF) or coupling faults (CF). The same-time standard march tests, which are based on only one run, are becoming insufficient for complex faults like pattern-sensitive faults (PSF). To increase fault coverage, the multi-run transparent march test algorithms have been used. This solution is especially suitable for built-in self-test (BIST) implementation. The transparent BIST approach presents the incomparable advantage of preserving the content of the random access memory (RAM) after testing. We do not need to save the memory content before the test session or to restore it at the end of the session. Therefore, these techniques are widely used in critical applications (medical electronics, railway control, avionics, telecommunications, etc.) for periodic testing in the field. Unfortunately, in many cases, there is very limited time for such test sessions. Taking into account the above limitations, we focus on short, two-run march test procedures based on counter address sequences. The advantage of this paper is that it defines requirements that must be taken into account in the address sequence selection process and presents a deeply analytical investigation of the optimal address decimation parameter. From the experiments we can conclude that the fault coverage of the test sessions generated according to the described method are higher than in the case of pseudorandom address sequences. Moreover, the benefit of this solution seems to be low hardware overhead in implementation of an address generator.

Keywords: RAM, BIST, PSF faults, march tests, multiple tests