DOI: 10.15199/13.2015.8.15

Kontrolowane generowanie optymalnych testów losowych

dr inż IRENEUSZ MROZEK, prof. dr hab. inż. VYACHESLAV YARMOLIK

Politechnika Białostocka, Wydział Informatyki

Złożoność współczesnych systemów elektronicznych stale rośnie. Dlatego bardzo ważną kwestią staje się ich efektywne testowanie. W przypadku systemów z ograniczoną liczbą wejść można używać testów wyczerpujących. Niestety wykładnicza zależność realizacji testu od liczby wejść systemu ogranicza zastosowanie tej techniki do szczególnych typów systemów charakteryzujących się relatywnie małą liczbą wejść lub oprogramowania z małym rozmiarem domeny danych wejściowych [1, 2].

Koncepcjami pozwalającymi częściowo ominąć restrykcje związane z liczbą wejść testowanego systemu są testy lokalnie wyczerpujące [3] lub pseudowyczerpujące [4–7]. Są one rzeczywistą alternatywą dla testów wyczerpujących. Pozwalają w sposób znaczny zredukować liczbę wektorów testowych. Jest to możliwe dzięki temu, iż często wszystkie wartości wyjściowe lub ich znaczna część zależne są od stosunkowo małego podzbioru danych wejściowych [1, 8].

Bardzo często dobrą aproksymacją testów wyczerpujących i testów pseudowyczerpujących są testy losowe [9–11]. Testowanie losowe jest procesem podawania losowych wartości na wejścia testowanego systemu. Z definicji, każdy wektor testowy jest generowany w sposób losowy niezależnie od wygenerowanych wcześniej wektorów. Jako zalety testowania losowego można wymienić niski koszt generowania wektorów testowych, łatwość implementacji, możliwość generowania dużej liczby wektorów testowych czy możliwość wygenerowania testów bez znajomości specyfikacji testowanego systemu. Dodatkowo technika ta wprowadza losowość w proces testowania.

Standardowe testy losowe nie wykorzystują jednak informacji jaka jest dostępna w środowisku testowania *czarnej skrzynki*. Dlatego można wykorzystać techniki kontrolowanego generowania testów losowych (ang. *Contrlled Random Tests*), które korzystając z powyższych informacji pozwalają zwiększyć efektywność procesu testowania losowego. Pierwszym podejściem do kontrolowanego testowania losowego jest testowanie antylosowe (ang. *Antirandom Testing*) [12]. W podejściu tym i podejściach pochodnych wykorzystywana jest określona miara i każdy kolejny wektor testowy generowany jest tak aby jego *odległość* od wszystkich wykorzystanych dotychczas wektorów była jak największa [13, 14].

Miary odległości wektorów testowych

Definicję kontrolowanego testu losowego można przedstawić w sposób następujący:

Definicja 1: Kontrolowany test losowy $T = \{T_0, T_1, T_2, \ldots, T_{q-1}\}$ jest testem z losowo wybranymi wektorami T_i, $i \in \{0,1,2,3,\ldots q\text{-}1\}$, gdzie T_i spełnia określone kryterium.

Jako miarę odległości pomiędzy wektorami testowymi często wykorzystywana jest odległość Hamminga lub standardowa odległość kartezjańska [12].

Definicja 2: Odległość Hamminga HD(T_i,T_j) pomiędzy dwoma wektorami binarnymi T_i, T_j wyraża liczbę pozycji na których wektory T_i i T_j różnią się i obliczana jest jako waga w(T_i,T_j) (liczba jedynek) wektora $T_i \oplus T_j$

$$HD(T_i,T_j) = w(T_i \oplus T_j) = \sum_{l=0}^{N-1}(t_{i,l} \oplus t_{j,l}). \tag{1}$$

Definicja 3: Odległość kartezjańska CD(T_i,T_j) pomiędzy dwoma wektorami binarnymi T_i,T_j dana jest jako:

$$CD(T_i,T_j) = \sqrt{(t_{i,0} - t_{j,0})^2 + \cdots + (t_{i,N-1} - t_{j,N-1})^2} = \sqrt{HD(T_i,T_j)}. \tag{2}$$

Przykładowo, w wypadku pary wektorów $A = (0000)$ i $B = (1010)$ HD(A,B) = 2 i CD(A,B) = $\sqrt{2}$. Generując kolejny wektor testowy w testach antylosowych wykorzystywane jest kryterium maksymalnej odległości kolejnego wektora względem wszystkich wektorów użytych dotychczas w procesie testowania. Technika ta ma na celu zwiększenie efektywności testowania poprzez uwzględnienie hipotezy zakładającej dużą liczbę wspólnych uszkodzeń wykrywanych przez wektory testowe będące w niewielkiej odległości od siebie i jednocześnie mniejszą liczbę wspólnych uszkodzeń w wypadku wektorów testowych znacznie oddalonych od siebie [12]. Dla procedur testowych złożonych z większej niż dwa liczby wektorów testowych zostały w [12] zaproponowane metryki uwzględniające odległości sumaryczne.

Definicja 4: Sumaryczna odległość Hamminga (THD), sumaryczna odległość kartezjańska (TCD) dla dowolnego wektora T_i jest sumą jego odległości Hamminga (kartezjańskich) w odniesieniu do wszystkich wcześniejszych wektorów testowych $T_0, T_1, \ldots, T_{i-1}$.

$$THD(T_i) = \sum_{j=0}^{i-1} HD(T_i,T_j) \tag{3}$$

$$TCD(T_i) = \sum_{j=0}^{i-1} CD(T_i,T_j) \tag{4}$$

Definicja 5: Test antylosowy z maksymalną odległością (ang. Maximal Distance Antirandom Test – MDAT) jest to test w którym każdy kolejny wzorzec T_i jest dobierany tak aby odległości sumaryczne THD lub TCD pomiędzy wektorem T_i i wektorami $T_0, T_1, T_2, \ldots, T_{i-1}$ były maksymalne.

Definicja 6: Sumaryczna odległość Hamminga testu T i sumaryczna odległość kartezjańska testu T dla dowolnego $T = \{T_0, T_1, T_2, \ldots, T_{q-1}\}$ jest sumą odległości Hamminga HD(T_i, T_j) i odpowiednio odległości kartezjańskich CD(T_i, T_j) dla wszystkich $i \neq j$; $i,j \in \{0,1,2,\ldots,q\text{-}1\}$.

$$THD(T) = \sum_{i=1}^{q-1} \sum_{j=0}^{i-1} HD(T_i,T_j) \tag{5}$$

$$TCD(T) = \sum_{i=1}^{q-1} \sum_{j=0}^{i-1} CD(T_i, T_j) \quad (6)$$

Rozpatrzmy antylosowy test T z czterema wektorami testowymi $q = 4$ każdy o szerokości $N = 3$ bity. Wygenerowany test antylosowy z maksymalną odległością będzie miał postać $T = \{000,111,010,101\}$ [12]. Niestety podstawowa metoda generowania testów antylosowych wymaga dużych nakładów obliczeniowych związanych z wyliczaniem odległości pomiędzy wszystkimi potencjalnymi wektorami z dziedziny wejściowej [12]. Nawet w wypadku poprawionych algorytmów złożoność obliczeniowa generowania testu antylosowego jest zbyt wysoka dla rzeczywistych rozmiarów N wektorów testowych. Dlatego szuka się dróg umożliwiających obniżenie złożoności obliczeniowej generowania kontrolowanych testów losowych.

W oparciu o przedstawione powyżej metryki zostanie skonstruowany optymalny kontrolowany test losowy (ang. *Optimal Controlled Random Test* – OCRT) charakteryzujący się maksymalną wartością powyższych metryk. Następnie zaproponowany zostanie algorytm o niskiej złożoności obliczeniowej umożliwiający generowanie testów OCRT.

Jako pierwszy wektor tworzonego testu OCRT wybrany może zostać dowolny N-bitowy wektor T_0 ze zbioru 2^N wszystkich możliwych wektorów testowych. Można przyjąć $T_0 = 000\ldots0$ co w żaden sposób nie zmniejsza ogólności prowadzonych rozważań [12]. Generując kolejny wektor testowy T_1 szukany jest wektor optymalny w stosunku do T_0. Zgodnie ze wszystkimi przedstawionymi powyżej metrykami wektorem takim jest wektor komplementarny do T_0. W rozpatrywanym przypadku $T_1 = \overline{T}_0 = 111 \ldots 1$. Wszystkie metryki osiągną maksymalną możliwą wartość: $HD(T_0,T_1) = THD(T_1) = THD(T) = N$, $CD(T_0,T_1) = TCD(T_1) = TCD(T) = \sqrt{N}$. Do otrzymania kolejnego wektora testowego T_2 nie jest możliwe wykorzystanie metryki $HD(T_i,T_j)$ i $CD(T_i,T_j)$, gdyż maksymalizacja ich wartości prowadzi do nieakceptowalnego wyniku $T_2 = T_0$. Do tych samych wniosków dochodzi się rozważając metryki $THD(T_i)$ i $THD(T)$. W tym przypadku dla testu $T = \{T_0,T_1,T_2\}$ gdzie $T_1 = \overline{T}_0$ dowolna wartość T_2 sprawia, iż $THD(T_2) = N$ i $THD(T) = 2N$.

Kandydatem na trzeci optymalny wektor testowy T_2 jest dowolny wektor spełniający warunek $T_2 \neq T_1 \neq T_0$. Rozpatrzmy wektor T_2 dla którego $HD(T_0,T_2) = Z$, wtedy $HD(T_1,T_2) = N-Z$. Sumaryczne odległości kartezjańskie będą równe odpowiednio: $TCD(T_2) = CD(T_0, T_2) + CD(T_1, T_2) = \sqrt{Z} + \sqrt{N-Z}$, $TCD(T) = CD(T_0, T_1) + CD(T_0, T_2) + CD(T_1, T_2) = \sqrt{N} + \sqrt{Z} + \sqrt{N-Z}$. Wtedy wartości maksymalne tych odległości uzyskamy dla $Z = N/2$ jako rozwiązanie równania $\delta(\sqrt{Z} + \sqrt{N-Z}/\ \delta Z = 0$. W dalszej analizie załóżmy, że N jest wartością parzystą. Zatem w rozpatrywanym przypadku pierwsze $N/2$ bitów wektora $T_2 = 00\ldots011\ldots1$ przybierze wartość 0, zaś druga połowa wartość 1. Wybierając kolejne wektory testu optymalnego musimy skupić się na tych których waga $w(T_i) = N/2$. Zapewni to maksymalną wartość $TCD(T_i)$ i $TCD(T)$. W wypadku wektora T_3 oczywistym wyborem jest zanegowana wartość wektora T_2, czyli $T_3 = \overline{T}_2 = 11\ldots100\ldots0$. Przy poszukiwaniu kolejnych wektorów testu optymalnego dążymy zatem do maksymalizacji wartości $TCD(T_i)$ i $TCD(T)$. W wypadku parzystego numeru wektora T_i, gdzie $i \in \{0,2,4,2k-2\}$ rozpatrywane metryki przybiorą odpowiednio wartości:

$$maxTCD(T_i) = i \times \sqrt{N/2}$$
$$maxTCD(T) = (i/2) \times \sqrt{N} + (i^2/2) \times \sqrt{N/2},\ i \in \{0,2,4,\ldots 2k-2\}.$$

W praktyce wektory parzyste są wybierane tak aby odległość Hamminga nowego wektora T_i od wszystkich poprzednich wektorów T_j była równa $HD(T_i,T_j) = N/2$ dla $j<i$. Wektory nieparzyste są generowane poprzez negację poprzedzającego ich wektora parzystego $T_i = \overline{T_{i-1}}$, $i \in \{1,3,5, \ldots,2k-1\}$. Dla wektorów wejściowych o szerokości $N = 2^m$ bitów liczba wektorów składających się na optymalny test kontrolowany $T = \{T_0, T_1,\ldots,T_{q-1}\}$ jest równa $2(m+1)$. Dla $m = 3$ test OCRT złożony jest z $2(m+1) = 2(3+1) = 8$ wektorów testowych, których wartości przedstawione są w *tabeli 1*.

Tab. 1. Optymalny test kontrolowany dla $N = 8$

Tabl. 1. Optimal controlled random test for $N = 8$

T_i	$t_{i,7}$	$t_{i,6}$	t_{i5}	$t_{i,4}$	$t_{i,3}$	$t_{i,2}$	$t_{i,1}$	$t_{i,0}$
T_0	0	0	0	0	0	0	0	0
T_1	1	1	1	1	1	1	1	1
T_2	0	0	0	0	1	1	1	1
T_3	1	1	1	1	0	0	0	0
T_4	0	0	1	1	0	0	1	1
T_5	1	1	0	0	1	1	0	0
T_6	0	1	0	1	0	1	0	1
T_7	1	0	1	0	1	0	1	0

Dla przypadku uogólnionego liczba wzorców testu OCRT jest równa $q = (\lceil log_2 N \rceil + 1)$, a efektywny sposób ich generowania znaleźć można w [5].

Generowanie zbioru testów optymalnych

Na bazie podstawowego testu OCRT, którego postać przypomniana została w p. 2, zostanie obecnie zaprezentowany algorytm umożliwiający efektywne generowanie zbioru testów OCRT dla $N = 2^m$.

Na podstawie Definicji 2 można zauważyć, iż w wypadku dowolnej macierzy zmiana porządku jej kolumn nie wpływa na odległość Hamminga pomiędzy dwoma dowolnymi wierszami tej macierzy (w szczególności pomiędzy wzorcami testu). Fakt ten wykorzystany został w Algorytmie 1 umożliwiającym generowanie zbioru testów OCRT.

Algorytm 1

1. Zainicjowanie macierzy M. Macierz M złożona jest z N kolumn i $q = 2(m + 1)$ wierszy wypełnionych wartościami kolejnych wzorców testowych testu OCRT wygenerowanego zgodnie ze schematem przedstawionym w p. 2.
2. Wygenerowanie macierzy maskującej M^* . Macierz maskująca M^* otrzymywana jest poprzez zmianę porządku kolumn macierzy M. Do zmiany porządku kolumn może zostać użyty dowolny algorytm permutacji.
3. Wygenerowanie wzorca pierwotnego P. Jako wzorzec pierwotny P można przyjąć losowo wygenerowany N-bitowy wzorzec ze zbioru wszystkich możliwych 2^N wzorców. Wygenerowany wzorzec pierwotny można przyjąć jako pierwszy wzorzec testowy: $T_0 = P$.
4. Wygenerowanie kolejnego wzorca testowego. Każdy kolejny wzorzec T_i otrzymywany jest jako suma: $T_i = P \oplus M^*_i$.
5. Wygenerowanie pozostałych wektorów testu OCRT Pozostałe wektory testu generowane są poprzez powtarzanie kroku 4, aż do wygenerowania wszystkich $2(m + 1)$ wzorców.

Analiza przedstawionego algorytmu pozwala stwierdzić, iż umożliwia on w dość efektywny sposób wygenerować zbiór wektorów testu OCRT w zależności od wektora pierwotnego *P*. Dla ogólnego przypadku liczba wygenerowanych wektorów testowych testu OCRT jest równa $q = 2(\lceil log_2 N \rceil + 1)$. Należy jednak zauważyć, iż proste zmiany wzorca pierwotnego *P* (krok 3 algorytmu) lub macierzy maskującej *M** (krok 2 algorytmu) umożliwiają przy niskiej złożoności obliczeniowej generowanie kolejnych testów OCRT (krok 4 i 5 algorytmu) różnych od testów już wygenerowanych. Wydaje się, iż dla dowolnej wartości *N* najbardziej wymagającą czynnością jest wygenerowanie macierzy *M* i permutacja jej kolumn w celu otrzymania macierzy maskującej *M**. W rzeczywistości dwukrotne zwiększanie wartości *N* wymaga jedynie powiększenia macierzy *M* o dwa wiersze. Nowa macierz konstruowana jest poprzez rozszerzenie dotychczasowych *N*-bitowych wierszy macierzy *M* do rozmiaru 2*N*. Rozszerzenie to realizowane jest poprzez proste powielenie wszystkich bloków występujących w wektorach z tymi samymi wartościami 0 i 1 zaś dwa dodatkowe wiersze M_{2m-2} i M_{2m-1} przyjmują odpowiednio wartości 010101…01 i 101010…10. Zmiana porządku kolumn wydaje się nie zawsze konieczna i może być zrealizowana z użyciem algorytmu losowego o małej złożoności.

Podsumowanie

W artykule przeanalizowane zostały podstawowe metryki, znane z literatury przedmiotu, odnoszące się do pomiaru zróżnicowania generowanych wzorców testowych. W oparciu o zaprezentowane metryki skonstruowany został optymalny kontrolowany test losowy (ang. *Optimal Controlled Random Test* – OCRT) w którym wykorzystane metryki przyjmują najwyższe możliwe wartości. Na bazie skonstruowanego testu OCRT zaproponowany został algorytm umożliwiający łatwe generowanie kolejnych testów OCRT charakteryzujących się tymi samymi właściwościami co pierwotnie skonstruowany test. Największą zaletą proponowanego algorytmu jest jego niska złożoność obliczeniowa.

Artykuł powstał w ramach realizacji pracy statutowej S/WI/3/13na Wydziale Informatyki Politechniki Białostockiej.

Literatura

[1] Nagle, H., Roy, S., Hawkins, C., McNamer, M. and Fritzemeier, R. Design for testability and built-in self test: a review, Industrial Electronics, IEEE Transactions on 36(2), 1989, 129–140.

[2] McCaffrey, J. D. An empirical study of the effectiveness of partial antirandom testing, Proceedings of the 18th International Conference on Software Engineering and Data Engineering, SEDE, 2009, pp. 260–265.

[3] Furuya, K. A probabilistic approach to locally exhaustive testing, Trans. of IEICE (Transactions of the Institute of electronics, information and communication engineers) E72(5), 1989, pp. 656–660.

[4] Kuhn, R. D. and Okum, V. Pseudo-exhaustive testing for software, Proceedings of the 30th Annual IEEE/NASA Software Engineering Workshop, IEEE Computer Society, Washington, DC, USA, 2006, pp. 153–158.

[5] Das, D. and Karpovsky, M. Exhaustive and nearexhaustive memory testing techniques and their BIST implementations., Journal of Electronic Testing 10(3), 1997, 215–229.

[6] Gupta, S., Rajski, J. and Tyszer, J. Arithmetic additive generators of pseudo-exhaustive test patterns, IEEE Transactions on Computers 45, 1996, 939–949.

[7] Yarmolik, S. Iterative near pseudoexhaustive random testing, Informatics 2(26), 2010, pp. 66–75.

[8] Wang, L. and McCluskey, E. Circuits for pseudoexhaustive test pattern generation, IEEE Transactions on CAD of Integrated Circuits and Systems, 1988, pp. 1068–1080.

[9] Grindal, M., Offutt, J. and Andler, S. F. Combination testing strategies – a survey, Technical Report ISE-TR-04-05, 2004, GMU Technical Report.

[10] Malaiya, Y. K. and Yang, S. The coverage problem for random testing, Proceedings of the 1994 IEEE International Test Conference, ITC '84, 1984, pp. 237–245.

[11] Shahbazi, A., Tappenden, A. F. and Miller, J. Centroidal voronoi tessellations – a new approach to random testing, IEEE Transactions on Software Engineering 39(2), 2013, pp. 163–183.

[12] Malaiya, Y. K. Antirandom testing: Getting the most out of blackbox testing, Proceedings of 6th IEEE International Symposium on Software Reliability Engineering, ISSRE '95, IEEE Computer Society, 1995, pp. 86–95.

[13] Mrozek, I. and Yarmolik, V. N. Antirandom test vectors for BIST in hardware/software systems, Fundam. Inform., 119(2), 2012, 163–185.

[14] Xu, S. and Chen, J. Maximum distance testing, w Asian Test Symposium, 2002, pp. 15–20.