Politechnika Poznańska

Wydział Elektroniki i Telekomunikacji

Justyna Zawada

On new class of test points and their applications

Autoreferat rozprawy doktorskiej

Promotor:
prof. dr hab. inż. Jerzy Tyszer

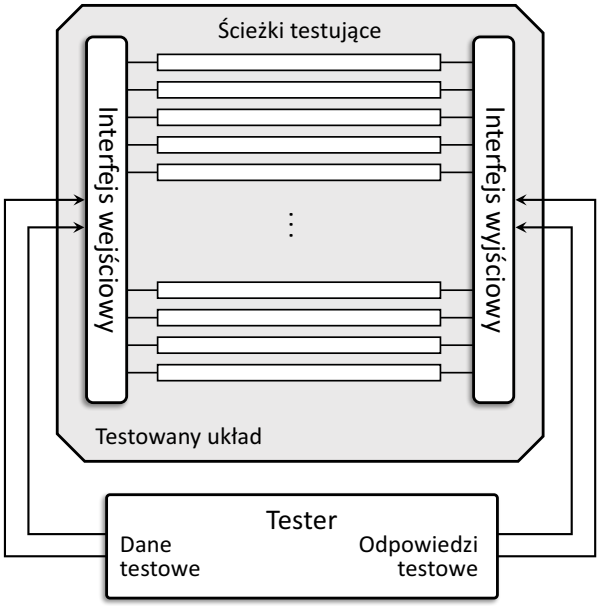
Poznań, 2017

1. Wprowadzenie

Bezprecedensowy rozwój techniki cyfrowej, w tym technologii wytwarzania cyfrowych układów scalonych, umożliwia produkcję coraz wydajniejszych urządzeń elektronicznych znajdujących zastosowanie w praktycznie każdej dziedzinie życia codziennego. Jak zaobserwował Gordon E. Moore [13], złożoność układów scalonych podwaja się średnio co dwa lata. Tak dynamiczny rozwój wiąże się z rosnącym zapotrzebowaniem na nowe rozwiązania pozwalające efektywniej testować współczesne układy wielkiej skali integracji. Wysokie wymagania stawiane nowym technologiom testowania wynikają także z coraz większego udziału układów cyfrowych w systemach o krytycznym znaczeniu dla zdrowia i bezpieczeństwa.

Niezawodność urządzeń elektronicznych jest w niekwestionowany sposób zależna od wysokiej jakości testowania produkcyjnego. Testowanie cyfrowych układów kombinacyjnych polega na podaniu na wejścia kolejnych pobudzeń i obserwowaniu wyjść celem potwierdzenia poprawności odpowiedzi. W przypadku układów sekwencyjnych test jest bardziej złożony z powodu znacznie większej przestrzeni stanów. Wraz ze wzrostem liczby elementów pamięci proces przeszukiwania stanów w celu pobudzenia i propagacji uszkodzeń staje się zadaniem niezwykle czasochłonnym (w praktyce niewykonalnym). Jednym z przełomowych rozwiązań w dziedzinie testowania układów cyfrowych umożliwiającym efektywne testowanie złożonych układów sekwencyjnych było wprowadzenie *ścieżki testującej*. Połączone szeregowo elementy pamięci układu tworzą rejestry przesuwające (Rys. 1.1) dostępne z zewnątrz w momencie przejścia w tryb testowy. Po wprowadzeniu pobudzeń, układ jest przełączany w tryb funkcjonalny celem zarejestrowania odpowiedzi, która następnie jest wysuwana z układu. Pomimo, iż ścieżki testujące stały się nieodzownym elementem większości cyfrowych układów scalonych, jest to metoda obarczona długim czasem niezbędnym dla wprowadzania danych testowych. Operacja ta wymaga n cykli zegara, gdzie n jest długością najdłuższej ścieżki testującej. W rezultacie proces właściwej aplikacji testów zajmuje niewielki procent całkowitego czasu testowania.

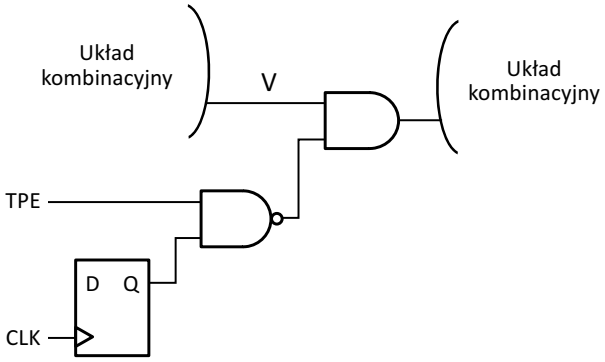
Postępująca miniaturyzacja elementów elektronicznych oraz gęste upakowanie komponentów w pojedynczym układzie znacząco zwiększa prawdopodobieństwo wystąpienia uszkodzeń. Współczesna bramka tranzystora to struktura o szerokości mierzonej w dziesiątkach atomów. Przy tak wysokiej precyzji nawet nieznaczne niedoskonałości procesu fabrykacji mogą prowadzić do przekroczenia marginesów błędów gwarantujących prawidłową pracę układu. Ponadto, nowoczesne techno-



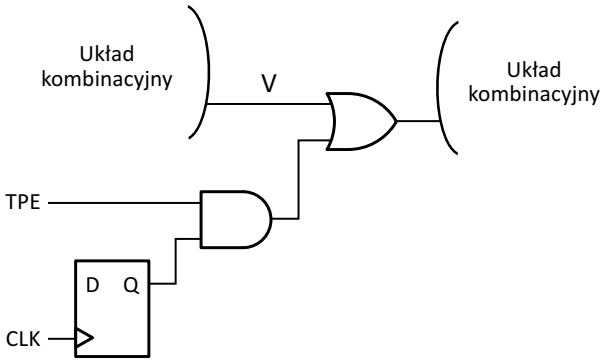
Rysunek 1.1: Testowanie z wykorzystaniem ścieżki testującej

logie produkcji wprowadzają nowe typy defektów, które w wielu przypadkach pozostają niewykrywalne dla klasycznych metod testowania. Wykorzystanie bardziej złożonych modeli uszkodzeń, uwzględniających na przykład strukturę układu na poziomie tranzystorów [5], jest z reguły okupione dużą liczbą dodatkowych pobudzeń, a zatem także wydłużeniem czasu testowania, nierzadko poza akceptowalne granice.

Urządzenia elektroniczne stosowane między innymi w medycynie, obronności, w systemach bezpieczeństwa transportu lotniczego, kolejowego lub samochodowego wymagają najwyższej jakości testu przez cały okres eksploatacji. Rozwiązaniem pozwalającym regularnie kontrolować sprawność układu w docelowym systemie jest autotestowanie, z reguły wykorzystujące wektory pseudolosowe (ang. logic built-in self-test, LBIST). W wielu przypadkach takie pobudzenia testowe uniemożliwiają osiągnięcie wystarczającego pokrycia uszkodzeń. Jedną z najważniejszych metod poprawiających efektywność wbudowanego testu są *punkty testowe* [6] zwiększające sterowalność (punkty kontrolne) i obserwowalność (punkty obserwacyjne) wewnętrznych węzłów w układzie. Punkty kontrolne (Rys. 1.2) pozwalają wymusić na danym węźle wartość logiczną, której prawdopodobieństwo wystąpienia w normalnych warunkach jest bardzo niskie. Na przykład, aby uzyskać wartość 1 na wyjściu 32-wejściowej bramki AND, wszystkie wejścia muszą przyjąć wartość 1. Prawdopodobieństwo wystąpienia takiej sytuacji wynosi tylko 2^{-32}. Wstawienie punktu kontrolnego typu OR (Rys. 1.2b) na wyjściu bramki AND znacząco poprawia sterowalność tej linii i może poprawić wykrywalność pewnych uszkodzeń.



(a) Punkt kontrolny typu AND



(b) Punkt kontrolny typu OR

Rysunek 1.2: Typy punktów kontrolnych

Istotnym i nowym problemem współczesnego testowania są próby wykorzystania elementów testujących do nielegalnego zidentyfikowania wewnętrznej struktury lub funkcjonalności układu. Zdobyta w ten sposób wiedza może być wykorzystana między innymi w procesie opracowywania własnych produktów lub do tworzenia nielegalnych kopii urządzeń elektronicznych. Jedną z metod mających na celu ochronę układu scalonego przed niepożądanym lub nieautoryzowanym dostępem jest maskowanie jego funkcjonalności (ang. logic locking) za pomocą dedykowanych elementów logicznych (ang. key gates), których poszczególne wejścia reprezentują bity klucza, zaś pozostałe są podłączone do wewnętrznych węzłów układu. Urządzenie działa prawidłowo po podaniu poprawnej sekwencji odblokowującej. Dotychczasowe metody maskowania funkcjonalności układu opierają się na dodawaniu nowych elementów, na przykład bramek typu XOR. Takie podejście obarczone jest dużym kosztem związanym między innymi z zajmowaną powierzchnią oraz możliwym obniżeniem szybkości układu.

W rozprawie zaproponowano nowe metody testowania układów cyfrowych oparte

na wykorzystaniu punktów testowych, które, jak wykazano, mogą ułatwić rozwiązanie wielu z zasygnalizowanych wyżej problemów. W tym celu opracowano oryginalne algorytmy wstawiania punktów testowych umożliwiających kolejno:

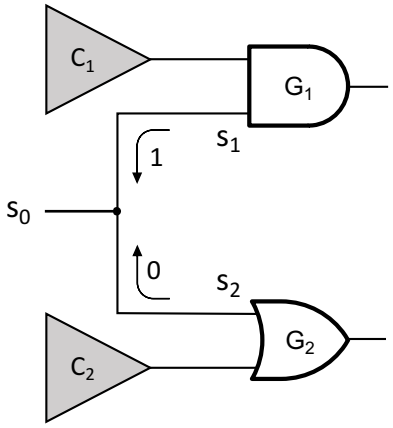
- redukowanie liczby testów deterministycznych,
- zwiększenie wydajności hybrydowej technologii łączącej testowanie deterministyczne z autotestowaniem losowym,
- skrócenie czasu testowania w układach z autotestem.

Ponadto, przedstawiono rozwiązanie wykorzystujące punkty testowe do maskowania funkcjonalności układu w przypadku niepożądanego dostępu. Zaprezentowana metoda w unikalny sposób zwiększa zarówno testowalność jak i bezpieczeństwo układów cyfrowych przy użyciu tych samych elementów infrastruktury testującej.

Możliwość zastosowania wszystkich oryginalnych rozwiązań potwierdzono eksperymentalnie wykorzystując wytwarzane obecnie na skalę przemysłową układy scalone wielkiej skali integracji. Oprogramowanie towarzyszące zaproponowanym w rozprawie metodom zostało zintegrowane z narzędziami komercyjnymi udostępnionymi przez firmę Mentor Graphics.

2. Redukcja liczby testów

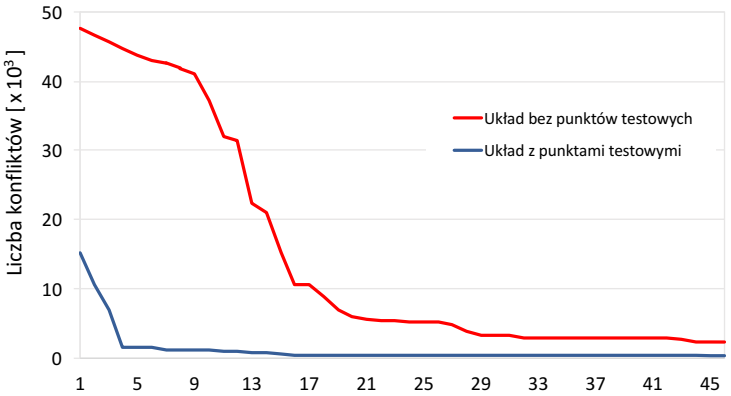
Wraz ze wzrostem złożoności układów cyfrowych, w nieunikniony sposób rośnie także liczba testów niezbędnych do osiągnięcia zadowalającego pokrycia uszkodzeń. W rozdziale 3. rozprawy zaproponowano metodę projektowania układów łatwo testowalnych redukującą liczbę testów poprzez zastosowanie nowego algorytmu rozmieszczania punktów testowych. W odróżnieniu od konwencjonalnych rozwiązań, charakterystyczną cechą przedstawionego podejścia jest wskazanie miejsc, w których algorytmy automatycznej generacji testów wykrywają konflikty przypisań między wartościami logicznymi niezbędnymi dla wykrycia uszkodzeń określanych jako uszkodzenia niekompatybilne. Przedstawiona metoda pozwala zredukować wielkość testu, a zatem skrócić także łączny czas testowania, w drodze wprowadzenia do układu dodatkowych punktów testowych eliminujących powyższe konflikty.



Rysunek 2.1: Konflikt przypisań

Rys. 2.1 ilustruje klasyczny konflikt przypisań. Przyjmijmy, że fragmenty układu C_1 oraz C_2 nie mają rozpływów wewnętrznych. W celu przepropagowania uszkodzeń z bloku C_1, wejście s_1 bramki G_1 musi przyjąć stan 1. Analogicznie, uszkodzenia z obszaru C_2 wymagają wartości 0 na linii s_2, co prowadzi do konfliktu przypisań na rozgałęzieniu s_0. Jeżeli uszkodzenia z C_1 i C_2 wymagają odpowiednio T_1 i T_2 wektorów testowych, wynikowy zbiór testów będzie zatem obejmował $T_1 + T_2$ pobudzeń. Wstawienie odpowiedniego punktu kontrolnego na jednej z gałęzi rozpływu rozwiązuje konflikt. Jeżeli $T_1 \approx T_2$, to zbiór testów można zredukować nawet o połowę. Warto zauważyć, że tradycyjne metody wskazywania lokalizacji punktów testowych mogą przeoczyć tego typu konflikt. Metryki oparte na analizowaniu sterowalności i obserwowalności układu nie uwzględniają bowiem procesów zacho-

dzących podczas generowania testów deterministycznych. Na przykład nie biorą pod uwagę zależności strukturalnych między uszkodzeniami. Przyjmijmy, że linia s_0 jest bezpośrednio połączona z komórką pamięci ścieżki testującej. W testowaniu losowym prawdopodobieństwo ustawienia s_0 na wartość 0 lub 1 wynosi wówczas 50%, co z punktu widzenia konwencjonalnych metod alokacji punktów testowych oznacza idealną sterowalność. W rezultacie, konflikt przypisań na gałęziach s_1 i s_2 zostanie zignorowany. Z tego powodu, wpływ klasycznych punktów testowych na redukcję deterministycznych pobudzeń jest nieznaczny i trudny do przewidzenia (od 0% do 35%) [8].



Rysunek 2.2: Profil konfliktów przypisań

Szereg eksperymentów potwierdza, że konflikty przypisań są częstym zjawiskiem w złożonych układach cyfrowych. Rys. 2.2 przedstawia przykładowy profil konfliktów w rzeczywistym układzie. Dla każdego rozpływu obliczono liczbę zablokowanych uszkodzeń dla wartości 0 i 1. Miarą konfliktu jest tutaj mniejsza z tych dwóch wielkości. Oś x (Rys. 2.2) reprezentuje uporządkowane malejąco rozpływy z największym konfliktem. Jak łatwo zauważyć, wstawienie punktów testowych w odpowiednich miejscach układu znacząco redukuje liczbę konfliktów (linia niebieska). Warto zauważyć, że oś x reprezentuje kolejne numery porządkowe rozpływów, a zatem dane przedstawione dla krzywej niebieskiej i czerwonej mogą odpowiadać innym rozgałęzieniom w układzie.

W celu wskazania rozgałęzień obarczonych największymi konfliktami zaproponowano (rozdział 3) szereg formuł opisujących zależności między uszkodzeniami podczas obliczania testu. Wprowadzono także szczegółowy opis mechanizmu blokowania propagacji uszkodzeń oraz podano sposób wymiarowania konfliktów. Analizę eksperymentalną przeprowadzono wykorzystując szereg istniejących układów, których charakterystyki zebrano w Tabeli 2.1. Kolejne kolumny zawierają liczbę bramek

Tabela 2.1: Charakterystyki badanych układów

Układ	Liczba bramek	Liczba przerzutników testujących	Konfiguracja ścieżek testujących	Kanały EDT	Kompresja
D1	1.25M	42K	136 × 340	3, 3	45x
D2	1.98M	47K	136 × 363	3, 3	45x
D3	4.68M	145K	608 × 239	6, 6	101x
D4	0.65M	23K	250 × 96	8, 8	31x
D5	0.51M	42K	189 × 412	8, 8	23x
D6	2.01M	44K	136 × 356	8, 8	17x
D7	1.90M	73K	371 × 189	8, 8	46x
D8	3.60M	95K	474 × 189	8, 8	59x
D9	1.00M	33K	298 × 95	8, 8	37x
D10	0.68M	25K	136 × 356	8, 8	17x
D11	0.44M	308K	800 × 388	5, 5	160x
D12	2.10M	148K	140 × 1058	2, 2	70x
D13	0.48M	242K	921 × 269	36, 36	26x

logicznych, liczbę elementów testujących, konfigurację ścieżek testujących, liczbę kanałów testera oraz stopień kompresji danych testowych otrzymanych w środowisku EDT [16].

Wpływ nowych punktów testowych na redukcję liczby testów przedstawiono w Tabeli 2.2. Wyniki dotyczą modelu uszkodzeń uwzględniającego strukturę układu na poziomie tranzystorów [5]. W kolejnych kolumnach podano liczbę punktów testowych (stanowiącą w przybliżeniu 2% elementów pamięci), pokrycie uszkodzeń oraz odpowiadającą mu liczbę wektorów testowych w układzie bez punktów testowych. Badaniu poddano uszkodzenia, dla których odpowiedź jest rejestrowana w jednym (1C) albo dwóch (2C) cyklach zegara. W ostatniej kolumnie zebrano wynikową redukcję liczby pobudzeń otrzymaną po wstawieniu punktów testowych w miejscach wyznaczonych przez proponowany algorytm.

Nowa metoda rozmieszczania punktów testowych umożliwia średnio ponad trzykrotną redukcję testów dla modelu uszkodzeń na poziomie tranzystorów [5], co z kolei oznacza skrócenie czasu i obniżenie kosztu testowania. W rozprawie wykazano także, że podobny poziom redukcji danych testowych jest możliwy dla innych modeli uszkodzeń. Opracowane formuły mogą ponadto służyć do zmniejszenia powierzchni zajmowanej przez punkty testowe [9]. Ponieważ przedstawione rozwiązanie zasto-

sowano po raz pierwszy w technologii kompresji EDT, w dalszej części autoreferatu termin punkty testowe EDT odnosić się będzie do metody zaprezentowanej w rozdziale 3. rozprawy.

Tabela 2.2: Redukcja liczby testów

Układ	Punkty testowe	Pokrycie uszkodz. [%]	Liczba testów	Typ uszkodz.	Redukcja testów
D1	1,740	92.62	14,710	1C	8.21x
D2	1,600	97.21	10,209	1C	1.93x
D3	3,200	76.94	12,086	2C	2.15x
D4	600	91.08	16,338	1C	2.92x
D5	1,648	97.65	25,855	1C	1.30x
D6	900	98.19	6,335	1C	2.98x
D7	1,468	95.56	24,269	2C	4.38x
D8	1,900	95.20	20,270	2C	2.63x
D9	670	90.78	18,517	2C	4.42x
D10	494	90.69	16,640	1C	2.90x
D11	4,500	77.34	69,606	2C	2.00x
D12	2,962	93.54	78,869	2C	3.20x
D13	5,000	74.65	115,091	2C	3.90x

3. Hybrydowe punkty testowe

Rozdział 4. rozprawy prezentuje metodę identyfikacji punktów testowych w hybrydowej technologii łączącej testowanie deterministyczne z autotestowaniem. Zaproponowane rozwiązanie równolegle redukuje liczbę testów deterministycznych oraz zwiększa pokrycie uszkodzeń dla pobudzeń losowych. Przedstawiona metoda umożliwia osiągnięcie docelowych parametrów jakościowych testu przy pomocy znaczniej mniejszej liczby punktów testowych niż w przypadku zastosowania dwóch oddzielnych klas dedykowanych dla testów deterministycznych i losowych.

Jako punkt wyjścia można potraktować dane zebrane w Tabelach 3.1 i 3.2, które ilustrują:

- wpływ punktów testowych EDT na pokrycie uszkodzeń dla testów losowych,
- redukcję pobudzeń deterministycznych uzyskaną dzięki tradycyjnym punktom testowym.

Pierwsze dwie kolumny tabel podają liczbę bramek oraz liczbę przerzutników testujących. Trzecia kolumna Tabeli 3.1 zawiera pokrycie uszkodzeń otrzymane dla 10 000 wektorów losowych w układzie bez punktów testowych. Kolejne dwie kolumny prezentują liczbę wykorzystanych punktów testowych oraz uzyskane pokrycie uszkodzeń. Trzecia kolumna Tabeli 3.2 podaje liczbę wektorów testowych w układzie bez punktów testowych. Ostatnia kolumna pokazuje zredukowaną liczbę testów otrzymaną po wstawieniu konwencjonalnych punktów testowych, określanych w kolejnych częściach autoreferatu jako punkty testowe LBIST.

Tabela 3.1: Punkty testowe EDT vs. pokrycie uszkodzeń wektorów losowych

Układ bez punktów testowych			Układ z punktami testowymi	
Bramki	Przerzutniki testujące	Pokrycie uszkodz. [%]	Punkty testowe	Pokrycie uszkodz. [%]
1.5M	75K	79.31	2K	84.08
2.6M	154K	76.46	1.5K	83.91
3.6M	41K	81.43	1.1K	84.63
2.6M	150K	83.29	3K	85.79

Wyniki z Tabeli 3.1 wskazują na wzrost liczby wykrytych uszkodzeń dla testów losowych, niemniej jednak punkty testowe EDT nie gwarantują ich pełnego pokrycia. Z kolei, punkty testowe przeznaczone dla pobudzeń losowych albo nieznacznie

Tabela 3.2: Tradycyjne punkty testowe vs. redukcja testów deterministycznych

Bramki	Przerzutniki testujące	Liczba testów	Punkty testowe	Liczba testów
2.1M	148K	10,175	3K	6,335
2.2M	143K	23,089	2.3K	**23,777**
4.4M	308K	69,606	4.5K	48,905
1.2M	63K	8,539	1.2K	**10,759**
Układ bez punktów testowych			Układ z punktami testowymi	

redukują liczbę testów deterministycznych, albo prowadzą do zwiększenia zbioru testów (pogrubione wartości w ostatniej kolumnie Tabeli 3.2). Otrzymane rezultaty wskazują, że algorytmy wstawiania omawianych grup punktów testowych nie identyfikują konfliktów występujących w teście hybrydowym. Zastosowanie oddzielnych klas punktów testowych przeznaczonych dla odpowiednich faz testowania wiąże się natomiast ze wzrostem zajmowanej powierzchni, jak również z ewentualnym obniżeniem wydajności układu. Co więcej, niekompatybilne metody wstawiania punktów testowych mogą mieć negatywny wpływ na jakość testu.

W oparciu o powyższe obserwacje, zaproponowano metodę wskazywania miejsc w układzie, w których wstawienie punktów kontrolnych jest korzystne zarówno dla testów deterministycznych jak i autotestu. Przedstawiony w rozprawie algorytm łączy formuły wykorzystywane w analizie konfliktów powstających podczas wyznaczania testów deterministycznych z opisem sterowalności i obserwowalności wewnętrznych węzłów układu. Temu celowi służy także zidentyfikowane i opisane w rozprawie zjawisko konfliktu hybrydowego.

Efektywność proponowanego rozwiązania zweryfikowano eksperymentalnie wykorzystując układy opisane w Tabeli 3.3. W kolejnych kolumnach podano parametry badanych układów. W eksperymentach porównano wpływ różnych klas punktów testowych na redukcję liczby testów oraz zwiększenie pokrycia uszkodzeń dla testów losowych. Dla każdego układu z Tabeli 3.3 rozpatrywano następujące kategorie punktów testowych:

- punkty testowe EDT,
- punkty testowe LBIST,
- hybrydowe punkty testowe użyte łącznie z punktami testowymi EDT.

Ostatnie rozwiązanie obejmuje dwie konfiguracje oznaczone jako EDT/H oraz H/EDT. W konfiguracji EDT/H najpierw wstawiano punkty testowe EDT, a następnie punkty hybrydowe. Z kolei H/EDT najpierw przeprowadza analizę punktów kontrolnych

Tabela 3.3: Charakterystyki badanych układów

Układ	Bramki	Przerzutniki testujące	Konfiguracja ścieżek testujących	Kanały EDT	Kompresja EDT	Punkty kontrolne	Punkty obserwacyjne
D1	1.2M	72K	400 × 181	4, 4	100x	500	1.5K
D2	2.3M	252K	490 × 515	4, 4	123x	1.2K	3.8K
D3	1.5M	162K	330 × 491	3, 3	110x	680	2.52K
D4	3.3M	326K	400 × 814	4, 4	100x	2K	4K
D5	1.2M	85K	428 × 200	6, 6	72x	600	1.1K
D6	2.1M	143K	400 × 359	4, 4	100x	800	2.2K
D7	1M	57K	400 × 143	4, 4	100x	600	1.4K
D8	1.6M	144K	400 × 366	4, 4	100x	1K	1.8K

z rozdziału 4. rozprawy, a następnie wykorzystuje algorytm wstawiania punktów testowych, jak to pokazano w rozdziale 3. W obydwu konfiguracjach każda klasa obejmuje połowę całkowitej liczby punktów testowych. We wszystkich grupach podział na punkty kontrolne i obserwacyjne jest taki, jak to przedstawiono w ostatnich dwóch kolumnach Tabeli 3.3.

Pierwsze dwie kolumny Tabeli 3.4 to liczba testów oraz odpowiadające im pokrycie uszkodzeń w układach bez punktów testowych. Pozostałe kolumny opisują wpływ różnych grup punktów testowych na redukcję testów przy zachowaniu takiego samego poziomu pokrycia uszkodzeń jak w układach odniesienia. Jak widać, z wyjątkiem układu D2, połączenie proponowanej metody z punktami testowymi EDT daje co najmniej taką samą redukcję pobudzeń testowych jak w przypadku metody dedykowanej. Zarówno dla konfiguracji EDT/H jak i H/EDT uzyskano średnią redukcję testów na poziomie 2,3x.

Tabela 3.4: Redukcja liczby testów

Układ	Liczba testów	Pokrycie uszkodz. [%]	EDT	EDT/H	H/EDT
D1	8,633	96.99	5,178	4,781	4,757
D2	4,397	99.06	1,805	2,048	1,966
D3	4,946	99.05	2,432	2,432	2,581
D4	3,515	96.49	2,150	2,066	2,117
D5	4,377	98.27	3,072	2,964	2,778
D6	24,027	99.51	8,429	7,817	7,478
D7	18,450	95.75	5,512	4,672	4,298
D8	2,874	97.60	1,803	1,471	1,551

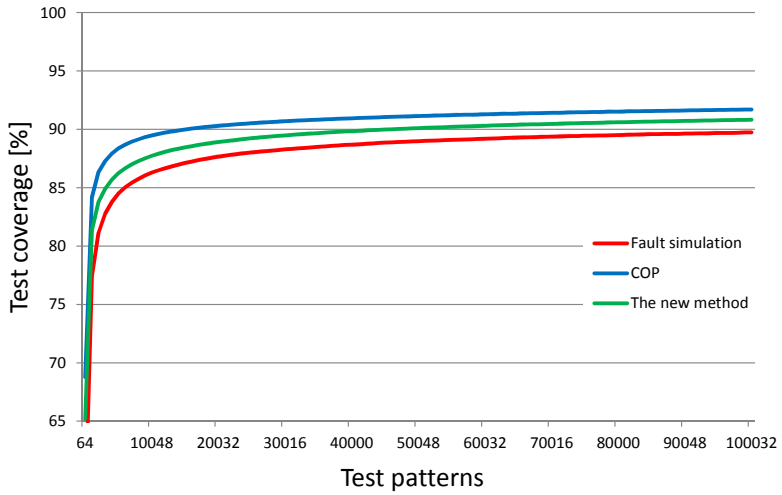
Kolejna grupa eksperymentów (Tabela 3.5) polegała na wyznaczeniu pokrycia uszkodzeń dla pobudzeń losowych w obecności różnych klas punktów testowych. Pierwsza kolumna to procent wykrytych uszkodzeń w układach bez punktów testowych, po podaniu 10 000 wektorów losowych. Kolejne kolumny prezentują pokrycie uszkodzeń po dodaniu odpowiednich grup punktów testowych. Warto zauważyć, że konfiguracje EDT/H oraz H/EDT reprezentują te same punkty testowe, dla których przedstawiono wyniki w Tabeli 3.4. Proponowane rozwiązanie ponownie oferuje wyższe pokrycie uszkodzeń w porównaniu z konwencjonalnymi punktami testowymi (z wyjątkiem układu D5, gdzie obserwujemy nieznaczne obniżenie wyniku na poziomie 0.15%).

W rozprawie zaproponowano także nowy i szybki algorytm szacowania pokrycia

Tabela 3.5: Pokrycie uszkodzeń

Układ	Pokrycie uszkodz.[%]	LBIST [%]	EDT/H [%]	H/EDT [%]
D1	76.70	82.75	87.70	87.88
D2	85.50	93.16	95.54	96.26
D3	82.82	86.24	93.03	92.26
D4	83.76	86.36	90.60	90.08
D5	85.08	90.82	90.67	90.65
D6	78.65	79.84	86.30	85.95
D7	74.53	82.32	89.69	89.43
D8	87.82	91.95	94.53	94.27

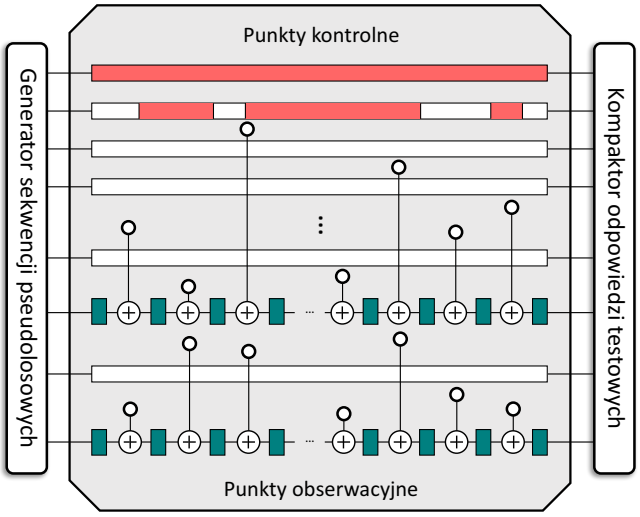
uszkodzeń dla potrzeb monitorowania procesu wstawiania kolejnych punktów testowych. Rys. 3.1 ilustruje procent wykrytych uszkodzeń w funkcji liczby testów dla układu D6. Linia czerwona reprezentuje rzeczywiste pokrycie uszkodzeń uzyskane dla 100 000 testów. Krzywe niebieska i zielona ilustrują wyniki otrzymane odpowiednio za pomocą klasycznego algorytmu COP [4] oraz metody przedstawionej w rozprawie. Jak widać, nowe podejście oferuje dokładniejsze oszacowanie pokrycia uszkodzeń. Podobne charakterystyki otrzymano także dla pozostałych układów.



Rysunek 3.1: Pokrycie uszkodzeń (układ D6, EDT/H)

4. Autotest z nowymi punktami testowymi

W rozdziale 5. rozprawy zaproponowano nowy sposób wykorzystania punktów testowych na potrzeby autotestowania. Opracowana metoda redukując czas testowania gwarantuje jednocześnie wysokie pokrycie uszkodzeń. Jest to możliwe dzięki rejestrowaniu odpowiedzi testowych za pomocą punktów obserwacyjnych w każdym cyklu zegara. Takie rozwiązanie stwarza warunki do podania zdecydowanie większej liczby wektorów testowych przy założonych ograniczeniach czasowych. Zaproponowane podejście jest szczególnie ważne dla układów o wyjątkowych wymaganiach niezawodnościowych, gdzie konieczne jest osiągnięcie wysokiego pokrycia uszkodzeń w bardzo krótkim czasie.



Rysunek 4.1: Autotest z nowymi punktami testowymi

Jak pokazano w rozdziale 2. rozprawy, metoda ścieżek testujących jest obarczona dużym kosztem związanym z czasem podawania testów. Wprowadzenie 10 000 sekwencji testowych do układu zawierającego ścieżki testujące o długości 100 przerzutników każda wymaga 1 miliona cykli zegara (100×10 000). Zarejestrowanie odpowiedzi układu trwa 10 000 cykli, co stanowi jedynie 1% czasu koniecznego dla transferu danych testowych. W rozprawie zaproponowano nowy sposób zastosowania punktów testowych w teście wbudowanym (Rys. 4.1) umożliwiający bardziej efektywne wykorzystanie czasu testowania. Zasadniczą różnicą między konwencjonalnym podejściem a omawianą metodą jest sposób podawania testów. Zmodyfikowana struktura ścieżek testujących stanowiących bazę dla punktów obserwacyjnych umożliwia rejestrowanie stanu układu w każdym takcie zegara. W kolejnych cyklach ścieżki

obserwujące przekazują kolejne bity wyniku do kompaktora odpowiedzi układu. Proponowana metoda pozwala podać n razy więcej testów niż klasyczny autotest, gdzie n jest liczbą przerzutników w najdłuższej ścieżce testującej. Pozostałe ścieżki testujące (na Rys. 4.1 zaznaczone na biało i czerwono) działają w konwencjonalny sposób.

Tabela 4.1: Charakterystyki badanych układów

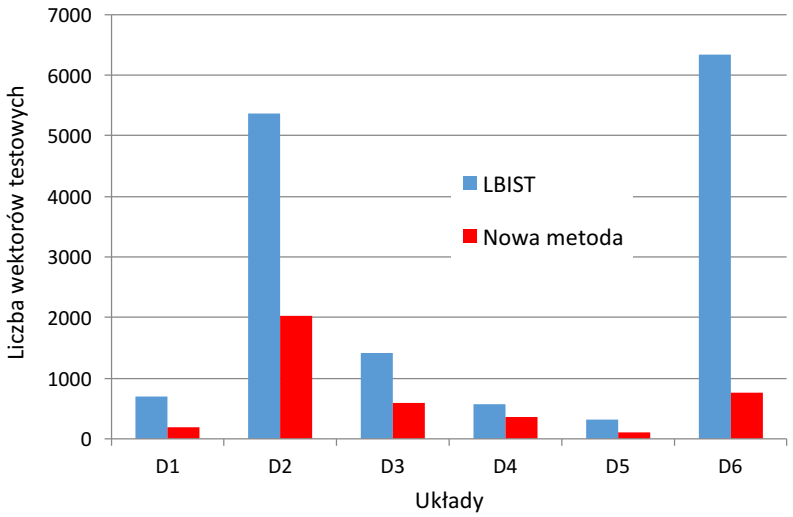
Układ	Bramki	Komórki testujące	Ścieżki testujące	Najdłuższa ścieżka testująca	Pokrycie uszkodz.	Liczba uszkodz. [%]
D1	453K	45K	226	200	81.07	1.6M
D2	1.21M	72K	382	190	78.53	4.5M
D3	1.19M	85K	427	200	86.54	4.1M
D4	2.62M	160K	1,015	158	90.34	9.2M
D5	1.62M	144K	700	207	88.83	4.3M
D6	372K	31K	54	647	73.33	1.1M

Nowe podejście wykorzystuje hybrydowe punkty testowe wprowadzone w rozdziale 4. rozprawy. Na potrzeby omawianego rozwiązania zaproponowano modyfikację algorytmu alokacji hybrydowych punktów testowych celem zwiększenia efektywności wykrywania uszkodzeń. Opracowane metody zostały zweryfikowane eksperymentalnie z użyciem układów przedstawionych w Tabeli 4.1. Ostatnie dwie kolumny tabeli reprezentują pokrycie uszkodzeń oraz ich całkowitą liczbę w układzie bez punktów testowych. Wyniki eksperymentu mającego na celu porównanie tradycyjnego autotestu stosującego punkty testowe z nowym podejściem zebrano w Tabeli 4.2. Pierwsze dwie kolumny przedstawiają podział punktów testowych. Kolejne sekcje odpowiadają pokryciu uszkodzeń uzyskanym przez konwencjonalny autotest (LBIST) oraz wzrost pokrycia (w stosunku do technologii LBIST) dzięki zastosowaniu nowej metody (Δ). Wyniki przedstawiono odpowiednio dla 1000 i 2000 wektorów testowych (w rozprawie dodatkowo dla 4000 i 10 000 testów). W rozprawie zamieszczono wykresy ilustrujące pokrycie uszkodzeń w funkcji czasu. Dla wszystkich badanych układów, prezentowane podejście osiąga wymagany i wysoki poziom pokrycia uszkodzeń znacznie szybciej niż tradycyjne rozwiązanie z punktami testowymi.

Tabela 4.2: Pokrycie uszkodzeń

Układ	Punkty kontrolne	Punkty obserwacyjne	1000 testów		2000 testów	
			LBIST [%]	Δ [%]	LBIST [%]	Δ [%]
D1	320	580	91.22	2.15	92.48	1.4
D2	700	800	85.14	2.98	87.74	2.25
D3	720	980	89.34	1.69	90.65	1.8
D4	1,000	2,000	93.06	1.29	95.12	0.73
D5	1,225	1,275	93.41	1.48	94.62	0.89
D6	1,019	780	81.13	10.08	85.09	8.26

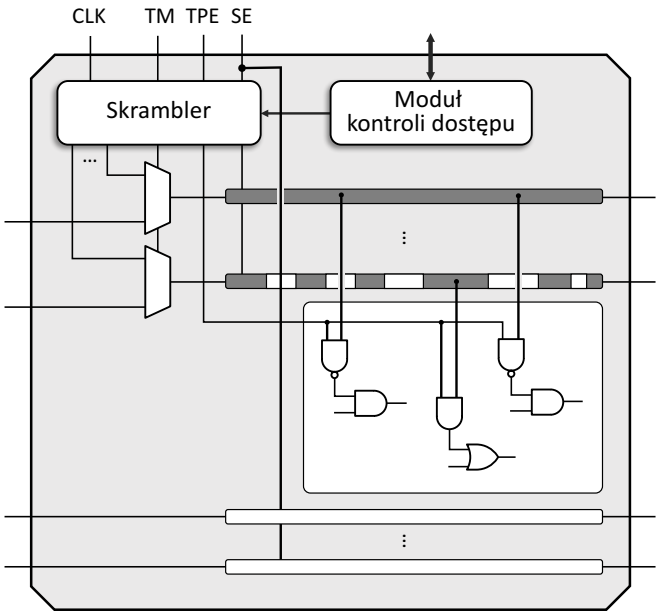
Rys. 4.2 ilustruje liczbę pobudzeń testowych wymaganą przez test wbudowany oraz proponowane rozwiązanie do wykrycia uszkodzeń na poziomie 90%. Typowo nowe podejście uzyskuje docelowe pokrycie trzykrotnie szybciej niż autotest konwencjonalny.



Rysunek 4.2: Pokrycie uszkodzeń na poziomie 90%

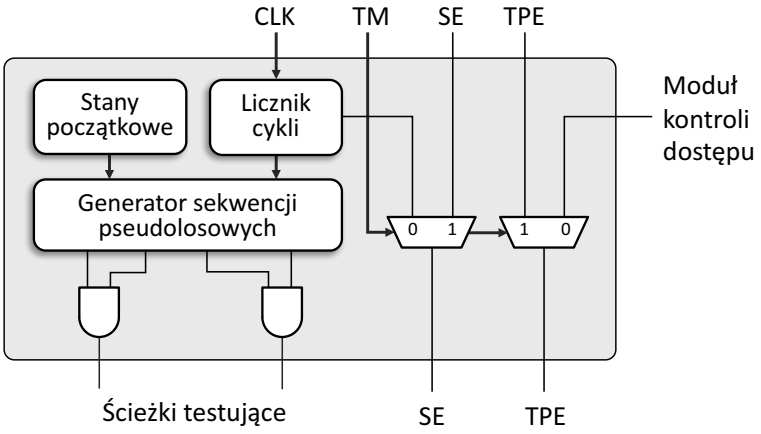
5. Ochrona układu cyfrowego

W ostatniej części rozprawy wykazano, że infrastruktura sprzętowa wprowadzona na potrzeby testu może zostać wykorzystana w trybie funkcjonalnym do poprawienia odporności układu na nielegalne próby zidentyfikowania jego wewnętrznej struktury i/lub funkcjonalności. Zaproponowane rozwiązanie wykorzystuje punkty testowe do celowego maskowania pracy układu cyfrowego w przypadku niepożądanego lub nieautoryzowanego dostępu. Efektywnie zatem przedstawiona metoda poprawia zarówno testowalność jak i bezpieczeństwo układów cyfrowych.



Rysunek 5.1: Zaproponowany sposób ochrony układu

W odróżnieniu od istniejących rozwiązań opartych na dodatkowych strukturach zmieniających funkcjonalność systemu, nowe podejście wykorzystuje istniejące w układzie punkty testowe (dotychczas transparentne w trybie funkcjonalnym). Zaproponowana metoda maskowania wprowadza dwa nowe elementy: moduł sterujący procesem ukrywania funkcjonalności (skrambler) oraz blok kontrolujący dostęp do układu (Rys. 5.1). Początkowo układ jest zablokowany, to znaczy, że wszystkie punkty testowe są włączone i tylko podanie odpowiedniego klucza wyłącza mechanizm ochronny układu. W przypadku aktywnego procesu maskowania funkcjonalności, skrambler dostarcza odpowiednie sekwencje binarne do ścieżek testujących, które sterują punktami kontrolnymi. W rezultacie, wybrane grupy punktów kontrolnych są cyklicznie uaktywniane i wyłączane, co z kolei zmienia sygnały w różnych częściach układu.



Rysunek 5.2: Proponowana realizacja skramblera

W rozprawie zaproponowano jedną z możliwych realizacji modułu sterującego procesem maskowania pracy układu (Rys. 5.2). W trybie testowym (aktywny sygnał TM), punkty kontrolne są uaktywniane przez tester, zaś w trybie funkcjonalnym (TM=0) przez skrambler. W momencie wykrycia niepożądanego dostępu, skrambler dostarcza do ścieżek testujących (aktywny sygnał SE) i dalej do punktów kontrolnych sekwencje losowe, które uaktywniają/wyłączają odpowiednie grupy punktów testowych. Szybkość zmian zależy od licznika cykli zegara. Każde uruchomienie urządzenia powoduje wprowadzenie generatora sekwencji losowych w nowy stan początkowy, co utrudnia analizowanie odpowiedzi układu. Takie postępowanie jest odpowiedzią na jeden z typowych ataków mających na celu rozpoznanie funkcjonalności zablokowanego układu jest podanie specjalnych sekwencji na wejścia i zarejestrowanie odpowiedzi. Następnie, te same pobudzenia wejściowe wprowadza się do układu z poprawnym kluczem. Wyniki porównania odpowiedzi mogą dostarczyć informacji umożliwiających całkowite lub częściowe zdekodowanie klucza.

Tabela 5.1: Charakterystyki badanych układów

Układ	Bramki	Wejścia pierwotne	Wyjścia pierwotne	Komórki testujące	Ścieżki testujące	TTPF [%]
D1	1.2M	536	528	85K	427	32
D2	453K	1,061	1,083	45K	226	19
D3	218K	2,364	237	14K	4	22
D4	1M	265	265	57K	48	23
D5	1.5M	2,829	3,069	75K	273	43
ethernet	108K	203	221	11K	106	21

Efektywność proponowanej metody zależy od liczby przerzutników oraz wyjść pierwotnych, do których punkty kontrolne mogą przesłać zakłóconą wartość logiczną. Warunkiem koniecznym jest istnienie ścieżki między punktem kontrolnym, a danym elementem układu. Osiągalność (pokrycie) wyjść pierwotnych oraz elementów pamięci zbadano dla układów w Tabeli 5.1. Ostatnia kolumna opisuje liczbę wstawionych punktów kontrolnych, wyrażoną jako procent całkowitej liczby punktów testowych. Wyniki pokrycia elementów pamięci oraz wyjść pierwotnych, wraz z odpowiadającą liczbą cykli zegara, zebrano w Tabeli 5.2.

Tabela 5.2: Pokrycie elementów pamięci

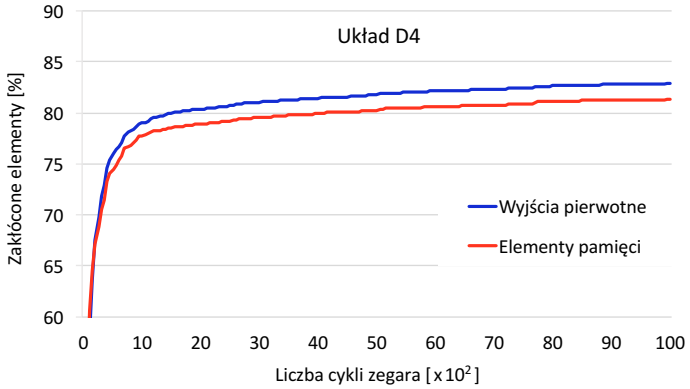
Układ	Wyjścia pierwotne		Przerzutniki testujące	
	Pokrycie elementów [%]	Liczba cykli zegara	Pokrycie elementów [%]	Liczba cykli zegara
D1	69.10	7	96.16	7
D2	81.39	11	77.23	12
D3	99.16	4	97.63	4
D4	99.57	4	84.87	5
D5	96.53	7	87.92	4
ethernet	73.20	5	95.00	9

Kolejne eksperymenty służyły zbadaniu rzeczywistego wpływu punktów kontrolnych na zakłócanie pracy układu. W tym celu, dla każdego układu z Tabeli 5.1 przeprowadzono symulację dwóch przypadków. Pierwszy zakłada, że układ działa poprawnie (podano prawidłowy klucz). Wszystkie elementy pamięci są inicjowane wartościami losowymi i następnie, przez 10 000 cykli zegara, układ jest pobudzany wartościami losowymi na wejściach pierwotnych. W każdym cyklu zegara dane ze ścieżek testujących oraz wyjść pierwotnych są rejestrowane i stanowią punkt odniesienia dla wyników otrzymanych po podaniu złego klucza. W przypadku aktywnego mechanizmu maskowania funkcjonalności, losowe grupy punktów kontrolnych są aktywowane w każdym cyklu zegara. Porównanie wyników zarejestrowanych w obydwu przypadkach podaje liczbę elementów pamięci oraz wyjść pierwotnych, które otrzymały różne wartości. Wyniki uśrednione dla 100 eksperymentów symulacyjnych zebrano w Tabeli 5.3. Każda kolumna to procent wszystkich wyjść pierwotnych oraz elementów pamięci posiadających co najmniej raz zakłóconą wartość logiczną w czasie 10 000 cykli zegara. Rys. 5.3 ilustruje ten proces w funkcji czasu. Rys. 5.4 pokazuje z kolei jaki procent elementów sekwencyjnych oraz wyjść pierwotnych za-

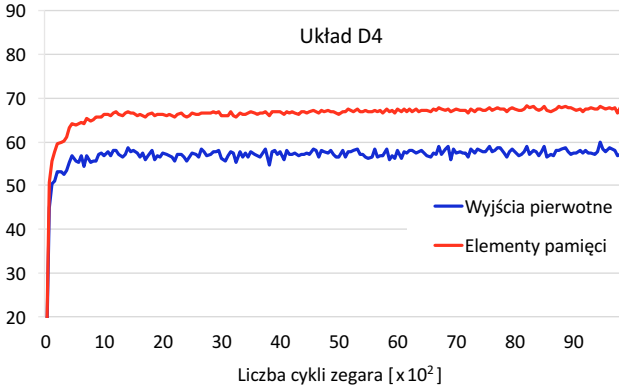
wiera przekłamaną wartość logiczną w każdym cyklu zegara. Prezentowane wykresy dotyczą układu D4. Szczegółowe wyniki dla pozostałych układów przedstawiono w rozdziale 6. rozprawy.

Tabela 5.3: Zakłócenia wartości logicznych po 10 000 cykli zegara

Zakłócenia [%]	D1	D2	D3	D4	D5	ethernet
Wyjścia pierwotne	20.65	79.81	87.98	25.49	60.25	45.52
Elementy pamięci	21.92	74.40	93.32	52.13	42.50	52.95



Rysunek 5.3: Zakłócenia wartości logicznych w elementach pamięci oraz wyjściach pierwotnych w funkcji czasu



Rysunek 5.4: Zakłócenia wartości logicznych w elementach pamięci oraz wyjściach pierwotnych w kolejnych cyklach zegara

6. Podsumowanie

W rozprawie zaproponowano metody testowania układów cyfrowych wielkiej skali integracji wykorzystujące nowe klasy punktów testowych wprowadzonych dla potrzeb redukcji liczby testów, zwiększenia wydajności hybrydowej technologii testowania, skrócenia czasu podawania pobudzeń testowych w urządzeniach z autotestem, oraz ochrony układy przed niepożądanym dostępem. W pierwszej części rozprawy zaproponowano nową metodę identyfikowania oraz rozwiązywania konfliktów przypisań między wartościami logicznymi niezbędnymi dla wykrycia grup uszkodzeń niekompatybilnych. Przedstawione podejście pozwala zredukować wielkość testu, a zatem skrócić także łączny czas testowania, w drodze wprowadzenia do układu punktów testowych w miejscach obarczonych największymi wartościami konfliktów. W kolejnej części rozprawy została opisana hybrydowa metoda identyfikacji punktów testowych. Proponowane rozwiązanie umożliwia osiągnięcie założonych parametrów jakościowych za pomocą znaczniej mniejszej liczby punktów testowych niż w przypadku zastosowania dwóch oddzielnych klas punktów testowych dedykowanych dla poszczególnych etapów testowania hybrydowego. Kolejna część rozprawy opisuje nowy sposób wykorzystania punktów testowych w teście wbudowanym znacząco skracający czas podawania pobudzeń losowych przy jednoczesnym zagwarantowaniu wysokiego pokrycia uszkodzeń. Rejestrowanie odpowiedzi układu za pomocą punktów obserwacyjnych w trakcie dostarczania pobudzeń testowych stwarza warunki do podawania zdecydowanie większej liczby testów niż było to możliwe dotychczas, przy założonych ograniczeniach czasowych. Zaproponowane podejście jest szczególnie ważne dla układów scalonych o podwyższonych wymaganiach niezawodnościowych wykorzystywanych na potrzeby przemysłu motoryzacyjnego, medycznego, obronnego, oraz lotniczego, gdzie krótki czas testowania z zachowaniem wysokiej jakości testu jest wymagany przez międzynarodowe normy bezpieczeństwa. W ostatniej części rozprawy wykazano, że punkty testowe pozwalają poprawić odporność układu na nielegalne próby zidentyfikowania jego wewnętrznej struktury i/lub funkcjonalności. Zaproponowane w rozprawie rozwiązanie wykorzystuje punkty testowe w trybie funkcjonalnym do celowego maskowania pracy układu w przypadku niepożądanego lub nieautoryzowanego dostępu. Przedstawiona metoda poprawia zarówno testowalność jak i bezpieczeństwo układów cyfrowych.

Wszystkie rozwiązania przedstawione w rozprawie zostały zweryfikowane i potwierdzone w trakcie obszernego programu badań eksperymentalnych przeprowadzonych z wykorzystaniem aktualnie produkowanych układów cyfrowych wielkiej skali

integracji oraz opracowanego przez autorkę oryginalnego oprogramowania będącego nietrywialnym rozszerzeniem istniejących narzędzi komercyjnych.

Bibliografia

[1] C. Acero, D. Feltham, F. Hapke, E. Moghaddam, N. Mukherjee, V. Neerkundar, M. Patyra, J. Rajski, J. Tyszer, and J. Zawada, "Embedded deterministic test points for compact cell-aware tests," in *Proc. ITC*, 2015, paper 2.2.

[2] C. Acero, D. Feltham, F. Hapke, E. Moghaddam, N. Mukherjee, V. Neerkundar, M. Patyra, J. Rajski, J. Tyszer, and J. Zawada, "On new test points for compact cell-aware tests," *IEEE Des. Test.*, vol. 33, no. 6, pp. 7–14, Dec. 2016.

[3] C. Acero, D. Feltham, Y. Liu, E. Moghaddam, N. Mukherjee, M. Patyra, J. Rajski, S. Reddy, J. Tyszer, and J. Zawada, "Embedded deterministic test points," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, 2017, accepted for publication.

[4] F. Brglez, P. Pownall, and R. Hum, "Applications of testability analysis: from ATPG to critical delay path tracing," in *Proc. ITC*, 1984, pp. 705–712.

[5] F. Hapke, W. Redemund, A. Glowatz, J. Rajski, M. Reese, M. Hustava, M. Keim, J. Schloeffel, and A. Fast, "Cell-aware test," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 33, pp. 396–1409, Sep. 2014.

[6] J. Hayes and A. Friedman, "Test point placement to simplify fault detection," *IEEE Trans. Comput.*, vol. 23, no. 7, pp. 73–78, Jul. 1974.

[7] H. Konuk, E. Moghaddam, N. Mukherjee, J. Rajski, D. Solanki, J. Tyszer, and J. Zawada, "Design for low test pattern counts," in *Proc. DAC*, 2015, paper 58.4.

[8] A. Kumar, J. Rajski, S. Reddy, and T. Rinderknecht, "On the generation of compact deterministic test sets for BIST ready designs," in *Proc. ATS*, 2013, pp. 201–206.

[9] Y. Liu, E. Moghaddam, N. Mukherjee, J. Reddy, S. M. Rajski, and J. Tyszer, "Minimal area test points for deterministic patterns," in *Proc. ITC*, 2016, paper 2.4.

[10] S. Milewski, N. Mukherjee, J. Rajski, J. Solecki, J. Tyszer, and J. Zawada, "Full-scan LBIST with capture-per-cycle hybrid test points," in *Proc. ITC*, 2017, accepted for publication.

[11] E. Moghaddam, N. Mukherjee, J. Rajski, J. Tyszer, and J. Zawada, "On test points enhancing hardware security," in *Proc. ATS*, 2016, pp. 61–66.

[12] E. Moghaddam, N. Mukherjee, J. Rajski, J. Tyszer, and J. Zawada, "Test point insertion in hybrid test compression/LBIST architectures," in *Proc. ITC*, 2016, paper 2.1.

[13] G. Moore, "Cramming more components onto integrated circuits," *Electronics*, vol. 38, no. 8, pp. 114–117, Apr. 1965.

[14] J. Rajski, E. Moghaddam, N. Mukherjee, J. Tyszer, and J. Zawada, "Test point insertion for low test pattern counts," U.S. Patent Application 20,160,109,517, Apr., 2016.

[15] J. Rajski, E. Moghaddam, N. Mukherjee, J. Tyszer, and J. Zawada, "Test point-enhanced hardware security," U.S. Patent Application 20,170,141,930, May. 18, 2017.

[16] J. Rajski, J. Tyszer, M. Kassab, and N. Mukherjee, "Embedded deterministic test," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 23, pp. 776–792, May 2004.