

AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE
WYDZIAŁ INFORMATYKI ELEKTRONIKI I TELEKOMUNIKACJI

KATEDRA ELEKTRONIKI



AUTOREFERAT ROZPRAWY DOKTORSKIEJ
TESTOWANIE RDZENI PROCESOROWYCH
IMPLEMENTOWANYCH W FPGA
TESTING OF SOFT PROCESSOR CORES
IMPLEMENTED IN FPGA

Autor: mgr inż. Mariusz Węgrzyn

Promotor: dr hab. inż. Ernest Jamro, prof. AGH
Promotor pomocniczy: dr. inż. Agnieszka Dąbrowska-Boruch

Kraków 2020

Motywacja

Podatność układów FPGA na zjawiska Single Event Upsets (SEU) ogranicza ich adaptację w systemach bezpieczeństwa i w aplikacjach stosowanych w krytycznych misjach np. kosmicznych. Zjawiska SEU najczęściej powodowane są przez cząsteczki promieniowania kosmicznego, przechodzące przez krzem, w wyniku którego indukowane są błędy w systemie. W przypadku układów FPGA, zjawisko SEU objawia się przede wszystkim zmianą pamięci konfiguracyjnej, w szczególności zawartości pamięci LUT. Testowanie FPGA wymaga innych rozwiązań od tych stosowanych dla układów scalonych Application Specific Integrated Circuit (ASIC). Znaczenie tych zjawisk zwiększa się wraz z zmniejszającymi się wymiarami technologii półprzewodnikowych oraz wraz z wysokością nad poziomem morza. Jest ono szczególnie uciążliwe w przypadku przestrzeni kosmicznej.

Rdzenie softprocesorowe, które reprezentują jedną z typowych aplikacji implementowanych w FPGA są również obiektem narażonym na pojawianie się błędów indukowanych poprzez SEU. Zatem w przypadku pracy systemu mikroprocesorowego w krytycznych aplikacjach, powinien on być okresowo testowany i rekonfigurowany, jeżeli błąd zostanie wykryty.

Cel i tezy pracy

Praca badawcza miała na celu opracowanie metodologii projektowania kompaktowego i efektywnego testu softprocesora zaimplementowanego w układach FPGA opartych na pamięciach SRAM. Zaproponowane rozwiązanie czerpie źródło z aplikacyjnie zorientowanego Built-In-Self-Test (BIST). Zaproponowałem metodologię opartą na tym, że sekwencja instrukcji assemblerowych powinna formować tzw. ścieżkę wrażliwą na dane. Zapożyczyłem to pojęcie z literatury, jednak w pracy ma ono inne nowatorskie znaczenie, bowiem ścieżka wrażliwa na dane jest oparta na bijektywności bloków programowych. Zasada ta ma za zadanie zapewnić możliwość generacji wszystkich możliwych pobudzeń (wektorów wejściowych) i doprowadzić do arbitralnych sytuacji, przy których trudne błędy zostaną wykryte. To rozwiązanie ma na celu uzyskanie przez program testowy maksymalnego pokrycia błędów, przy jednoczesnym zachowaniu jego kompaktowej architektury.

W związku z powyższymi potrzebami sformułowana została pierwsza teza pracy:

Teza1.

Stosując zasadę ścieżki wrażliwej na dane, która wykorzystuje bijektywną właściwość programu testowego, można znacząco ułatwić procedurę testową oraz poprawić pokrycie błędów.

Innym znaczącym osiągnięciem pracy jest zaproponowana metodologia wstrzykiwania błędów. W układach ASIC błędy najczęściej modeluje się poprzez

podłączenie ścieżki na trwałe do masy lub napięcia zasilania (stuck-at-0, stuck-at-1). W przypadku układów FPGA taka metodologia nie odzwierciedla w pełni zjawiska SUE w pamięci LUT. Dlatego zaproponowano nową metodologię, w której pojedynczy bit pamięci LUT ulega przekłamaniu. Dodatkowo aby zamodelować błędy połączeń (stuck-at) zmianie może ulec więcej niż jeden bit pamięci LUT.

Proces modelowania błędów został zautomatyzowany, dzięki czemu możliwe było przeprowadzenie symulacji wszystkich możliwych błędów SUE zgodnie z zaproponowaną metodologią. Dodatkowo proces symulacji procesora został przeprowadzony w trzech wymiarach. Oprócz symulacji różnych błędów SUE, następnym wymiarem jest symulowanie w pierwszym przybliżeniu wszystkich instrukcji assemblerowych – każda instrukcja stanowi niezależny blok bijektywny. Trzecim wymiarem jest symulowanie tych instrukcji dla różnych wektorów (danych) wejściowych. W konsekwencji powyższej symulacji została stworzona metodologia oraz algorytmy optymalizacyjne, które pozwoliły na redukcję liczby koniecznych do przetestowania wektorów bez wpływu na pokrycie błędów. Dodatkowo dla procesora PicoBlaze pokazano, że już przetestowanie trzech wektorów daje bliskie pełnego pokrycie błędów. W wyniku przeprowadzonych prac sformułowano drugą tezę pracy.

Teza2.

Heurystyki optymalizacyjne w połączeniu z proponowaną metodologią wstrzykiwania błędów mogą znacząco zredukować liczbę wektorów testowych wymaganych do osiągnięcia maksymalnego pokrycia błędów procesorów implementowanych w FPGA.

Organizacja dysertacji

Praca została podzielona na dziesięć rozdziałów.

Rozdział 2: „Radiation-induced errors in microelectronic circuits.” („Błędy indukowane przez promieniowanie w układach mikroelektronicznych.”) opisuje fizyczne mechanizmy generowania błędów typu Single Event Upset (SEUs) indukowanych przez promieniowanie w układach mikroelektronicznych. Błędy te są sklasyfikowane jako trzy rodzaje, w zależności od ich wpływu na funkcjonowanie FPGA. Następnie zilustrowana jest propagacja zjawisk SEU w układach kombinacyjnych i przerzutnikach. Pod koniec tego rozdziału rozważona jest manifestacja zjawisk SEU w komórkach pamięci RAM oraz podstawowych architekturach FPGA. Także kwestie związane z modelowaniem SEU w Look-up-Tabelach (LUT) są wprowadzone w tym rozdziale.

W rozdziale 3: „Software-based self-test of embedded processor cores.” („Softwarowy self-test wbudowanych rdzeni procesorowych.”) zawarte zostało wprowadzenie do tematu związanego z testowaniem rdzeni procesorowych. W tym rozdziale są opisane wybrane rozwiązania z bibliografii przedmiotu strukturalnego i funkcjonalnego testowania wbudowanych rdzeni procesorowych. Omawiane są zagadnienia rozwijania wzorców testowych dla każdego komponentu procesora w przypadku strukturalnych self-testów. Poruszane są tutaj ograniczenia dostępu do szczegółów architektonicznych procesorów z powodu ochrony własności intelektualnej i proponowane wówczas metody pseudolosowej generacji wzorców testowych. Alternatywne, deterministyczne strukturalne metodologie testowe są przybliżane w przypadku braku powyższych ograniczeń. Finalnie prezentowane są wybrane funkcjonalne metody self-testów bazujące na funkcjonalnym opisie instrukcji assemblerowych procesora.

Rozdział 4: „Fault Injection.” („Wstrzykiwanie błędów.”) zapoznaje czytającego z technikami wstrzykiwania błędów (Fault Injection - FI) stosowanymi do efektywnej ewaluacji rozwijanych metod testowych. Te metody są klasyfikowane jako bazujące na symulacjach lub eksperymentach. Obie klasyfikacje mogą być implementowane zarówno softwarowo jak i hardwarowo. Często wstrzykiwanie błędów stanowi techniczne wyzwanie i wymaga zaawansowanych dedykowanych projektów.

Rozdział 5: „Proposed Solution: Sensitive-Path Approach.” („Proponowane rozwiązanie: Ścieżka wrażliwa na dane”) W tym rozdziale zaproponowana została metodologia której celem jest kompozycja kompaktowej sekwencji testowej, wykrywającej permanentne SEU – indukowane błędy w rdzeniach procesorowych implementowanych w FPGA opartych na pamięciach SRAM. Rozwijane eksperymenty są zorientowane na maksymalne pokrycie błędów, osiągnięte dzięki nowej metodologii. Została zaproponowana nowa koncepcja „Ścieżki Wrażliwej na Dane”, która opiera się na bijektywności każdego testowanego bloku / instrukcji assemblerowej. Jest to jedno z głównych nowatorskich osiągnięć. Prezentowany jest w przykładach program testowy dla procesora PicoBlaze oparty na tej koncepcji. Zostały także opracowane i opisane ulepszenia bijektywnych bloków w celu osiągnięcia pełnego pokrycia błędów. W tym rozdziale prezentowane są również rezultaty eksperymentalne osiągnięte przez program testowy na różnych etapach jego rozwoju.

W Rozdziale 6: „Reduction of number of test vectors.” („Redukcja liczby wektorów testowych.”) zaprezentowane zostały heurystyki optymalizacyjne mające na celu redukcję liczby wektorów testowych. Znaczenie tych metod wzrasta istotnie w przypadku procesorów 32 lub 64 bitowych dla których przetestowanie wszystkich możliwych kombinacji wejściowych jest praktycznie niemożliwe. Zostały zaproponowane trzy algorytmy optymalizacyjne. 1) Najpierw wektory, które wykrywają największą liczbę błędów – „Algorytm zachłanny”, 2) Najpierw wektory które wykrywają najtrudniejsze do wykrycia błędy - „Algorytm najniższego rzędu”, 3) „Algorytm Hybrydowy”, który łączy cechy algorytmu pierwszego i drugiego. Również cykliczne użycie rezultatów jest proponowane w tym rozdziale. Polega ono na

używaniu rezultatów wykonania programu testowego, jako nowe wektory testowe. Autor zdeterminował optymalne zestawy globalnych oraz lokalnych zestawów wektorów testowych dla testowania całego procesora oraz jego indywidualnych bloków funkcjonalnych. Te eksperymenty wykazały znaczące różnice pomiędzy testowalnością poszczególnych bloków procesora.

Rozdział 7: „MicroBlaze case study.” („Przypadek studyjny MicroBlaze.”) opisuje eksperymenty z wybranym przez autora procesorem MicroBlaze. Została tutaj zaprezentowana idea implementacji pierwszego programu testowego i główne zasady jego kompozycji. Teoria ścieżki wrażliwej na dane została zastosowana tutaj, podobnie jak w przypadku kompozycji programu testowego dla PicoBlaze. Są tutaj wytłumaczone niektóre z przykładów początkowej wersji programu dla MicroBlaze. Następnie opisane są problemy związane z ewaluacją efektywności programu testowego. Pomimo tych problemów, są tutaj prezentowane pewne rezultaty badań, które mogą być porównywane z bibliografią przedmiotu. Finalnie rezultaty badań nad PicoBlaze zostały zastosowane do kompozycji programu testowego dla MicroBlaze i przykłady kodu w assemblerze są prezentowane pod koniec tego rozdziału.

Rozdział 8: „Evaluation of the test program.” („Ewaluacja programu testowego.”) prezentuje schemat ewaluacji programu testowego. Poddane dyskusji są tutaj takie tematy jak: projekt środowiska dla eksperymentów, proponowany przez autora nowatorski model błędów w Tabelach Look-up układów FPGA. Również proponowana jest technika i środowisko dla dedykowanego FPGA wstrzykiwania błędów. Opisane są również strukturalny VHDL PicoBlaze i skrypty pomocnicze.

Rozdział 9: „Problem of faults masking.” („Problem maskowania błędów.”) analizuje szczegółowo problem maskowania błędów. Ten problem jest rozważany odpowiednio dla każdego bloku mikroprocesora. Niniejszym niewykryte, pozostałe po wykonaniu programu testowego błędy są klasyfikowane do kilku kategorii. Są tutaj badane różne rodzaje redundancji logicznych i hardwarowych. Pewne mechanizmy wykrywania błędów są wytłumaczone na przykładach. Metody służące do rozwiązywania problemów maskowania błędów są proponowane w niektórych przypadkach.

Podsumowanie

Zaproponowano nową metodologię kompozycji kompaktowej sekwencji testowej, która wykrywa permanentne błędy SEU – indukowane w rdzeniach procesorowych implementowanych w FPGA bazujących na pamięciach SRAM. Moim nowatorskim założeniem jest, że sekwencja testowa reprezentuje ścieżkę wrażliwą na dane, jeżeli przepływ danych przez nią jest wrażliwy na zmiany wzorców testowych (**Wegrzyn 2009**).

Jedną z najważniejszych nowych idei tej pracy jest zaproponowana bijektywna właściwość programu testowego. Zaproponowałem metodę osiągnięcia bijektywności

na poziomie instrukcji assemblerowych. Program skomponowany z bijektywnych bloków osiągnął zdecydowanie lepsze pokrycie błędów (85,6%) niż znane aplikacje obliczeniowe lub programy testowe o prostszej architekturze. Zdecydowanie najlepsze pokrycie błędów (94,76 %) osiągnąłem kreując program złożony z lokalnych bijektywnych programów, które generują jednocześnie pełny cykl lokalnych wektorów testowych.

Jedną z najważniejszych nowości tutaj wprowadzonych jest nowy model wstrzykniętych błędów. Model funkcjonalny tych błędów różni się znacząco od konwencjonalnych modeli „stuck-at” w związku z tym, że SEU – indukowane błędy wpływają na konfiguracyjny SRAM, a zatem mogą zmienić logikę implementowaną przez Tabele Look-up (LUTs). Dlatego metody testowania oraz model błędów w FPGA różni się istotnie od znanych modeli „stuck-at” w ASIC-ach. Nikt przede mną nie wprowadził takiego modelu błędów według proponowanych zasad (**Wegrzyn 2009**), (**Wegrzyn 2014A**). Ja odzwierciedliłem SEU indukowane błędy w LUT-ach zgodnie z naturą zjawisk fizycznych w półprzewodnikach jak to opisano w bibliografii poświęconej badaniu tych zjawisk na poziomie fizyki (**Gaspard 2017**), (**Rebaudengo 2002A**), (**Rebaudengo 2002B**). Korzyści moich nowatorskich metod są podwójne, ponieważ w ten sposób można modelować zarówno błędy w LUT-ach, które prowadzą do innej od zamierzonej implementacji funkcji logicznych generowanych przez LUT-y. Należy zauważyć, że SEU – indukowane błędy są trudniejsze do wykrycia w porównaniu z błędami „stuck-at”, które są główną miarą pokrycia błędów stosowaną w literaturze.

Drugą korzyścią jest fakt, że te błędy mogą być interpretowane w szczególnych przypadkach jako „stuck-at-0” lub „stuck-at-1” na wejściach lub wyjściach LUT-ów. W ten sposób błędy sieci połączeń FPGA są również symulowane. Wraz z heurystykami optymalizacyjnymi redukuje to znacznie liczbę wektorów testowych wymaganych do osiągnięcia pełnego pokrycia błędów.

Korzyści nowego modelu dla wstrzykiwania błędów

- Odzwierciedlają błędy SEU – indukowane w LUTs zgodnie z naturą fizyczną tych zjawisk w półprzewodnikach jak opisano w bibliografii
- Modelowanie błędów bazuje na modyfikacjach funkcji logicznych implementowanych przez LUTs
- Modelowane są także błędy typu stuck-at na wejściach i wyjściach LUTs jako szczególna interpretacja błędów w LUTs
- Błędy wstrzykiwane są tylko na jedną pozycję bitową w pamięci LUT dla każdej z symulacji – to odzwierciedla prawidłowo rzeczywiste zjawiska SEUs i stanowi większe wyzwanie dla programu testowego
- Wierny model błędów w połączeniu z automatyczną symulacją każdego możliwego błędu pozwala ograniczyć znacząco liczbę wektorów testowych i czas testowania

Oryginalne osiągnięcia autora

- Wprowadzenie nowej bijektywnej metodologii testowania bazującej na zasadzie ścieżki wrażliwej na dane
- Znacząco wyższa efektywność osiągnięta przez program bijektywny niż inne aplikacje obliczeniowe np. mnożenie macierzy
- Rozwój oryginalnej bijektywnej metody na poziomie instrukcji assemblerowych.
- Kompaktowy rozmiar bijektywnego programu około 370 instrukcji assemblerowych
- Wykrycie, badania i rozwiązanie problemu braku pełnego cyklu
- Osiągnięcie zdecydowanie wyższego pokrycia błędów przez program złożony z lokalnych podprogramów, który generuje w pełni cykliczne rezultaty niż inne aplikacje
- Aktywowanie wysokiego procentu ścieżek wrażliwych na dane przez bijektywny w pełni cykliczny program testowy
- Opracowanie kompletnego systemu do testowania wybranych rdzeni procesorowych implementowanych w FPGA
- Metodologia implementacji skryptu w perlu, który umożliwia zautomatyzowane wstrzykiwanie błędów zgodnie z nowym modelem i steruje symulatorem CADENCE
- Rozwinięta efektywna cykliczna metodologia do testowania przemysłowego (w pamięci zapisany tylko wektor startowy i liczba iteracji)

Proponowane nowe heurystyki optymalizacyjne

- Dziewięciokrotnie zredukowana liczba globalnych wektorów testowych, co znacząco redukuje czas testowania
- Wykonanie programu testowego może być wielokrotnie szybsze niż operacja read-back
- Pojedynczy blok procesora wymaga średnio tylko 2 lokalne wektory do osiągnięcia FCmax (maksymalnego pokrycia błędów)
- FC powyżej 95% FCmax przy pierwszych 3-4 globalnych wektorach testowych

Dodatkowe wnioski z prac badawczych

- Osiągnięcie 100% FC przez program testowy napisany w assemblerze jest niemożliwe z powodu logicznych i hardwarowych redundancji
- Możliwa jest nieznaczna poprawa FC poprzez szczegółowe analizy interakcji pomiędzy danymi wejściowymi i logiką procesora
- Najtrudniejszym do testowania okazał się blok który generuje flagi ZERO i CARRY

Publikacje autora

Wegrzyn, M., Novak, F., Biasizzo, A. (2006). „*Application-oriented testing of FPGA circuits.*” 42rd International Conference on Microelectronics Devices and Materials MIDE M, Strunjan, Slovenia.

Wegrzyn, M., Novak, F., Biasizzo, A., Renovell, M.(2007 A). „*Functional test of processor cores in FPGA-based applications.*” Proc. Workshop on Electronic Testing 43rd International Conference on Microelectronics Devices and Materials MIDE M, Bled, 177-181.

Wegrzyn, M., Biasizzo, A., Novak, F. (2007 B). „*Application-oriented testing of embedded processor cores implemented in FPGA circuits.*” International Review on Computers and Software, 2: 666-671.

Wegrzyn, M., Biasizzo, A., Novak, F., Renovell, M., (2009). „*Functional Testing of Processor Cores in FPGA-Based Applications.*” Computing and Informatics, Vol.28, 2009, 97-113.

Wegrzyn, M., Sosnowski, J., (2014 A). „*Tracing Fault Effects in FPGA Systems*” INTERNATIONAL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS Vol.60, NO. 1, 103-108.

Wegrzyn, M., Sosnowski, J.,(2014 B). „*Testing schemes for systems based on FPGAs processor cores*” Pomiary, Automatyka, Kontrola Vol. 56, nr 01, 483-485.