



Zachodniopomorski  
Uniwersytet  
Technologiczny  
w Szczecinie

Wpłynęło dnia 26.02.2021  
Zarejestrowano pod nr .....  
Podpis ..... *[Signature]*



Prof. dr hab. inż. Aleksandr Cariow,  
Katedra Architektury Komputerów  
i Telekomunikacji  
Wydział Informatyki  
Zachodniopomorski Uniwersytet  
Technologiczny w Szczecinie  
ul. Żołnierska 52

Szczecin, dnia 22.02.2021 r.

**Recenzja rozprawy doktorskiej**  
**dla Rady Dyscypliny Automatyka, Elektronika i Elektrotechnika**  
**Akademii Górniczo-Hutniczej**

**Tytuł rozprawy:**

*Testing of soft processor cores implemented in FPGA*

**Autor rozprawy:**

mgr inż. Mariusz Węgrzyn

**Promotor:**

dr hab. inż. Ernest Jamro, prof. AGH

**Promotor pomocniczy:**

dr. inż. Agnieszka Dąbrowska-Boruch

Podstawę formalną wykonania recenzji stanowi uchwała Rady Dyscypliny Automatyka, Elektronika i Elektrotechnika Akademii Górniczo-Hutniczej im. Stanisława Staszica w Krakowie, z dnia 14 stycznia 2021 r. oraz pismo Przewodniczącego Rady dr hab. inż. Ryszarda Sroki, prof. Uczelni z dnia 18. 01. 2021r.

## **1. Aktualność tematu rozprawy**

Szybki rozwój technologii układów scalonych, gwałtowny wzrost złożoności strukturalnej i funkcjonalnej powodują ciągłą potrzebę rozwijania i doskonalenia metod testowych. Niekorzystny wpływ na funkcjonowanie elementów półprzewodnikowych i układów scalonych może wywołać oddziaływanie promieniowania jonizujące, którego wynikiem są różne skutki, związane z uszkodzeniami funkcjonalnymi lub katastroficznymi. Systemy elektroniczne są niezwykle czułe zarówno na długotrwałe działanie promieniowania, jak i pojedyncze uderzenia wysokoenergetycznych cząstek. Badania prowadzone w przemyśle pokazały, że promieniowanie takie może nie tylko wywołać nieniszczące zmiany stanów komórek pamięci, ale także spowodować trwałe uszkodzenia elementów układów elektronicznych. Wszystkie uszkodzenia wywoływane promieniowaniem są określane angielską nazwą *single event effects (SEE)*. Są dzielone na cztery rodzaje: *single event upsets (SEU)*, powodujące przemijające zmiany stanów logicznych; *single event latchups (SEL)*, powodujące powstawanie dużych prądów pasożytniczych w złączach *pnpn* w strukturach CMOS — prądy te mogą mieć charakter samopodtrzymujący się i mogą powodować zarówno uszkodzenia przemijające jak i trwałe; *single event burnout (SEB)*, powodujące trwałe uszkodzenia tranzystorów; *single event functional interups (SEFI)*, powodujące trwałe bądź przemijające wyłączenie funkcjonalności układu elektronicznego. Recenzowana praca została skoncentrowana wokół problemu zapobiegania szkodliwemu wpływowi efektu SEU, powodującemu przemijające zmiany stanów logicznych w układach scalonych. Istotny problem stanowi fakt, że w wyniku tego efektu powstają uszkodzenia polegające na zmianie stanu logicznego w układach FPGA wywołane przez cząstkę jonizującą, powodujące tzw. błąd miękki (*soft error*), bez defektu fizycznego, dający się naprawić, często zaś układ „nie widzi” tego typu uszkodzenia. Znaczenie SEU wzrasta kiedy technologiczne rozmiary struktur półprzewodnikowych maleją oraz ze względu na coraz bardziej złożone architektury układów. Narażeniom na promieniowanie jonizujące podlegają elementy układy scalone używane w aparaturze elektronicznej wykorzystywanej m.in. w satelitach prowadzących badania w kosmosie, satelitach telekomunikacyjnych. Wpływ SEU powodowanego promieniowaniem kosmicznym wzrasta wraz z wysokością. Testowanie, diagnozowanie błędów i rekonfiguracja FPGA są niezbędne w przypadku badań kosmicznych, które dzięki między innymi technice układów programowalnych rozwijają się bardziej dynamicznie w ostatnich latach i planowane są coraz większe wyzwania w tym obszarze. Obecnie produkowane układy FPGA odporne na promieniowanie są zwykle zbyt drogie, a także posiadają niewystarczające zasoby do implementacji bardziej złożonych projektów. Dlatego badania w oznaczonym kierunku są ciągle aktualne.

## **2. Zagadnienia naukowe rozpatrywane w pracy, cele oraz tezy rozprawy**

**Problemem naukowym, wokół którego** została skoncentrowana recenzowana praca jest testowanie FPGA aplikacyjne i strukturalne.

**Przedmiot badań** stanowią procedury testowania układów FPGA aplikacyjnego i strukturalnego oraz metody ewaluacji programów testowych poprzez wstrzykiwanie błędów.

**Głównym celem pracy** jest propozycja oryginalnego i jednocześnie praktycznego podejścia do testowania rdzeni procesorowych implementowanych w układach FPGA, pozwalającego na wygenerowanie kompaktowej sekwencji testowej, która wykrywa permanentne błędy SEU – indukowane w rdzeniach procesorowych implementowanych w układach FPGA bazujących na SRAM. W pracy m.in. podjęto próbę wygenerowania zwartego i wydajnego testu funkcjonalnego wbudowanych rdzeni procesorów zaimplementowanych w układach FPGA opartych na SRAM.

Nowatorskim założeniem jest to, że sekwencja testowa reprezentuje ścieżkę wrażliwą na dane, jeżeli przepływ danych przez nią jest wrażliwy na zmiany wzorców testowych. Na bazie tego nowatorskiego założenia i metodologii testowej, zostały sformułowane dwie tezy pracy.

**Teza 1:** Używając zasady wrażliwej ścieżki, która wykorzystuje własność bijektywną programu testowego może znacząco uprościć procedurę testową i poprawić pokrycie błędów.

**Teza 2:** 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.

### **3. Najgłówne zadania, które zostały rozwiązane w pracy**

1. Zaproponowano autorskie podejście do projektowania kompaktowego i efektywnego testu „softprocesora” zaimplementowanego w układach FPGA opartych na pamięciach SRAM.
2. Opracowano kompaktowy bijektywny program na poziomie instrukcji assemblerowych.
3. Zweryfikowana programistycznie bijektywność każdego z bloków i całego programu.
4. Osiągnięto wyższe pokrycie błędów przez program złożony z lokalnych podprogramów, który generuje w pełni cykliczne rezultaty niż inne aplikacje.
5. Opracowano kompletny system do testowania wybranych rdzeni procesorowych implementowanych w FPGA.

### **4. Analiza układu i struktury pracy**

Rozprawa jest zorganizowana w następujący sposób: w **Rozdziale 1** przedstawiono uzasadnienie wyboru tematu, celu i tezy pracy oraz omówiono jej układ.

**Rozdział 2** opisuje mechanizmy generowania błędów indukowanych przez promieniowanie w układach mikroelektronicznych. Błędy te są sklasyfikowane jako trzy rodzaje, w zależności od ich wpływu na funkcjonowanie FPGA. Ten rozdział prezentuje fizyczne mechanizmy zjawisk Single Event Upset. 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. Zostały też omówione różne rodzaje błędów SEU indukowanych w FPGA oraz kwestie związane z modelowaniem SEU w LUTach.

**Rozdział 3** zawiera wprowadzenie do tematu związanego z testowaniem rdzeni procesorowych. Zostały m.in. opisane niektóre rozwiązania z bibliografii przedmiotu strukturalnego i funkcjonalnego testowania wbudowanych rdzeni procesorowych. Przy strukturalnych self-testach, wzorce testowe są rozwijane dla każdego komponentu procesora, na bazie net-listy na poziomie bramek logicznych dla indywidualnych komponentów rdzenia. Ponieważ szczegóły na poziomie bramek rdzenia procesorowego w większości przypadków z powodu ochrony własności intelektualnej są nieosiągalne dla projektanta wzorce testowe są generowane w sposób pseudolosowy. Natomiast, kiedy informacja na poziomie bramek o rdzeniu procesorowym jest dostępna, może być zastosowana deterministyczna metodologia testowa i deterministyczne wzorce testowe mogą być generowane przez Automatyczny Generator Wzorców Testowych. W czasie funkcjonalnego self-testu rdzenie procesorowe są testowane poprzez wykonywanie sekwencji instrukcji, która sprawdza funkcjonalne zachowanie procesora. Projekt tego funkcjonalnego self-testu jest związany z funkcjonalnym opisem instrukcji procesora.

**Rozdział 4** zapoznaje z technikami wstrzykiwania błędów (Fault Injection (FI)) stosowanymi do efektywnej ewaluacji i waluacji rozwijanych metod testowych. Te metody są klasyfikowane jako bazujące na symulacjach lub eksperymentalne. Obydwa te rodzaje mogą być implementowane zarówno programowo jak i sprzętowo. Często wstrzykiwanie błędów stanowi techniczne wyzwanie i wymaga zaawansowanych dedykowanych projektów.

**W rozdziale 5** Autor zaproponował podejście, którego celem jest wygenerowanie kompaktowej sekwencji testowej, wykrywającej permanentne SEU – indukowane błędy w rdzeniach procesorowych implementowanych w FPGA opartych na SRAM. Rozwijane eksperymenty są zorientowane na maksymalne pokrycie błędów, osiągnięte dzięki jak to tylko możliwe kompaktowej architekturze programu testowego. W tym rozdziale Autor wprowadza nową koncepcję „ścieżki wrażliwej na dane” która opiera się na bijektywności każdego testowanego bloku/instrukcji assemblerowej. Zostały również opisane opracowane przez Autora programy testowe dla dwóch mikroprocesorów: PicoBlaze i MicroBlaze. Całą początkową ideą była kreacja ścieżki wrażliwej na dane poprzez znakomite właściwości programu assemblerowego, który zachowuje wszystkie dane. Zostały także opracowane i opisane ulepszenia bijektywnych bloków w celu osiągnięcia pełnego pokrycia błędów. Również zostały udostępnione eksperymentalne rezultaty, osiągnięte przez program testowy na różnych etapach jego rozwoju.

**W Rozdziale 6** Autor prezentuje heurystyki optymalizacyjne mające na celu redukcję liczby wektorów testowych. (Rozważając testowanie większych procesorów). Znaczenie tych metod wzrasta istotnie w przypadku bardziej wyrafinowanych procesorów. Trzy algorytmy optymalizacyjne zostały rozwinięte: „Najpierw wektory które wykrywają największą liczbę błędów – „Algorytm zachłanny”, „Najpierw wektory które wykrywają najtrudniejsze do wykrycia błędy”, „Algorytm Hybrydowy” który łączy cechy algorytmu pierwszego i drugiego. Również cykliczne użycie rezultatów jest proponowane w tym rozdziale. Autor zdeterminował optymalne zestawy globalnych oraz lokalnych zestawów wektorów testowych dla testowania całego hardware procesora oraz indywidualnych bloków funkcjonalnych. Te eksperymenty zostały zaprojektowane do optymalizacji testowania indywidualnych bloków procesora, kiedy taka potrzeba się pojawi, oraz kiedy zainteresowanie projektanta koncentruje się na

poszczególnych blokach np. podczas procesu projektowania. Te eksperymenty wykazały znaczące różnice pomiędzy testowalnością poszczególnych bloków procesora.

Podczas prac nad cyklicznym użyciem rezultatów pojawił się problem braku pełnego cyklu. Autor rozwiązał ten problem poprzez zastosowanie rejestru przesuwającego z liniowym sprzężeniem zwrotnym (LFSR). Rezultaty ulepszeń programu testowego dla PicoBlaze przy użyciu LFSR dla wszystkich instrukcji typu SHIFT są prezentowane w tym rozdziale.

**Rozdział 7** opisuje eksperymenty z wybranym przez Autora procesorem MicroBlaze. Idea implementacji pierwszego programu testowego i główne zasady funkcjonowania zostały tutaj zaprezentowane. Teoria ścieżki wrażliwej na dane została zastosowana w tym, podobnie jak w przypadku kompozycji programu testowego dla PicoBlaze. Niektóre z przykładów początkowej wersji programu dla MicroBlaze zostały wytłumaczone. Dalej zostały opisane problemy związane z ewaluacją efektywności programu testowego. Zaprezentowane również pewne rezultaty badań, które mogą być porównywane z rezultatami opisanymi w cytowanych przez Autora źródłach. 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** prezentuje schemat ewaluacji programu testowego. Poddane dyskusji tutaj są takie tematy jak: projekt środowiska dla eksperymentów, proponowana technika i środowisko dla dedykowanego FPGA wstrzykiwania błędów. Opisane są również strukturalny VHDL PicoBlaze i skrypty pomocnicze.

**Rozdział 9** analizuje szczegółowo problem maskowania błędów. Ten problem jest rozważany odpowiednio dla każdego bloku mikroprocesora. Niewykryte, pozostałe po wykonaniu programu testowego błędy są klasyfikowane do kilku kategorii. Różne rodzaje redundancji logicznych i hardwarowych są badane. Pewne mechanizmy wykrywania błędów są wytłumaczone na przykładach. Metody służące do rozwiązywania problemów maskowania błędów są proponowane w niektórych przypadkach.

**Rozdziałem 10** jest podsumowanie.

Praca składa się z 10 rozdziałów (w tym podsumowanie), 4 załączniki i zawiera 177 stron, 65 rysunków, 25 tabel oraz 30 listingów z przykładami programów testowych.

## **5. *Poprawność redakcyjna rozprawy***

Poprawność redakcyjna rozprawy nie budzi większych zastrzeżeń, chociaż w tekście rozprawy występują nieliczne uchybienia o charakterze redakcyjnym. Dotyczy to przeważnie nieuważnego zastosowania interpunkcji oraz przedimków „a” oraz „the”.

## **6. *Uwagi i komentarze recenzenta***

1. W rozprawie bardzo pobieżnie odniesiono się do możliwości wykrycia błędów SUE poprzez procedurę odczytu zapisanej uprzednio konfiguracji układu FPGA. Natomiast z treści wynika, że daje ona teoretycznie 100% pewność wykrycia błędu SUE i może być bardzo atrakcyjna w przypadku częściowej rekonfiguracji, dla której rozmiar odczytywanych danych może być wielokrotnie mniejszy. Prosiłbym o ustosunkowanie się do tej kwestii.

2. W pracy pokazano, że zastosowanie lokalnych wektorów testowych pozwala na znaczną redukcję czasu wykonywania programu testowego. Najwięcej miejsca poświęcono optymalizacji liczby wektorów i wartości wektorów globalnych. Czy podobną metodologię można zastosować do wektorów lokalnych lub też wielkości lokalnych bloków programu testującego?

3. Pewnym mankamentem jest to, że wybrane do testowania obiekty (MicroBlaze i PicoBlaze) obecnie są już trochę zdezaktualizowane, aczkolwiek nie przeszkadza to zrozumieniu i weryfikacji szczegółów proponowanych przez Autora rozwiązań i trików testowania.

4. Biorąc pod uwagę długoletni okres aktywności zawodowej doktoranta (lata 2005-2021) pewien niedosyt pozostawia relatywnie skromny ilościowo i nieco przedawniony dorobek publikacyjny doktoranta (tylko 6 publikacji, z których ostatnia publikacja ukazała się w roku 2014), jednak istotnym i raczej przekonywującym dla mnie atutem jest obecność w tym dorobku dwóch prac z listy A.

## **7. Konkluzja recenzji**

W podsumowaniu należy stwierdzić, że cel pracy, postawiony doktorantem, został osiągnięty. Praca świadczy o tym, że Autor dobrze widzi wszystkie niuanse studiowanego problemu naukowo-technicznego oraz potrafi przedstawić jego rozwiązanie. Uzyskane wyniki potwierdzają praktyczną przydatność i udowadniają postawione tezy i teoretyczne założenia.

Rezultaty zawarte w rozprawie mogą również stanowić punkt wyjścia do dalszych badań w zakresie studiowania i implementacji metod testowych.

Uważam, iż praca doktorska mgr inż. Mariusza Węgrzyna pod tytułem „Testing of soft processor cores implemented in FPGA” spełnia wymagania stawiane rozprawom doktorskim w dziedzinie nauk inżynieryjno-technicznych i Dyscyplinie Automatyka, Elektronika i Elektrotechnika w świetle obowiązujących przepisów, określonych w ustawie Prawo o szkolnictwie wyższym i nauce. W związku z tym wnoszę o dopuszczenie Pana mgr inż. Mariusza Węgrzyna do publicznej obrony.

