

dr hab. inż. Adam Milik, prof. PŚ.
Politechnika Śląska
Wydział Automatyki Elektroniki i Informatyki
Katedra Systemów Cyfrowych
ul. Akademicka 16
44-100 Gliwice

Gliwice, 6.03.2021

S E K R E T A R I A T
Rady Dyscypliny AEE

Wpłynęło dnia 10.03.2021
Zarejestrowano pod nr
Podpis *dm*

**Recenzja rozprawy doktorskiej mgr. inż. Mariusza Węgrzyna
dla Rady Dyscypliny Automatyka Elektronika i Elektrotechnika
Akademii Górniczo-Hutniczej im. Stanisława Staszica w Krakowie**

(podstawą opracowania recenzji jest uchwała Rady Dyscypliny
Automatyka, Elektronika i Elektrotechnika z dnia 14.01.2021)

Tytuł rozprawy: Testing of Soft Processor Cores Implemented in FPGA

Autor rozprawy: mgr inż. Mariusz Węgrzyn

Promotor rozprawy: dr hab. inż. Ernest Jamro, prof. AGH

Promotor pomocniczy: dr inż. Agnieszka Dąbrowska-Boruch

Dziedzina: nauki techniczne

Dyscyplina: Automatyka, Elektronika i Elektrotechnika

1. Zagadnienia naukowe rozprawy – cel i teza pracy

Układy FPGA są powszechnie wykorzystywane jako programowalne elementy logiczne do konstrukcji dedykowanych układów obliczeniowych i sterujących. W celu skrócenia czasu projektowania urządzeń cyfrowych, wykorzystuje się gotowe rdzenie złożonych układów logicznych. Bardzo chętnie wykorzystywane są rdzenie mikroprocesorów. Ich budowa została dostosowana do odwzorowania w strukturach logicznych układów programowalnych. Jednostki takie są oferowane jako rozwiązania dedykowane np. picoBlaze oraz microBlaze (Xilinx) lub NIOS II (Intel uprzednio Altera). Dostępne są również rozwiązania ogólnego przeznaczenia np. LatticeMico 8 i 32, które dostarczone są w postaci opisu syntezowalnego na poziomie przesłań międzyrejestrów.

W układach FPGA zastosowano statyczną pamięć RAM do przechowywania konfiguracji, dzięki czemu liczba przeprogramowań układu nie jest ograniczona. W układach narażonych na oddziaływanie cząsteczek jonizujących mogą wystąpić przejściowe impulsy wywołane zderzeniem cząsteczki z odpowiednim fragmentem struktury półprzewodnikowej. Powoduje to zaburzenie działania o charakterze impulsowym. W przypadku komórek pamięci

statycznej RAM może prowadzić to do niezamierzonej modyfikacji jej zawartości. Układy FPGA, które przechowują konfigurację w statycznej pamięci RAM narażone są na niezamierzoną zmianę konfiguracji wywołaną przez cząsteczki jonizujące. Modyfikacja taka może objawić się poprzez zmianę zawartości bloków tablicowych LUT lub modyfikację sieci połączeń, prowadząc do nieoczekiwanego działania osadzonego w układzie programowalnym systemu cyfrowego.

Praca doktorska mgr. inż. Mariusza Węgrzyna odpowiada na potrzeby związane z rozwojem metod diagnostyki układów implementowanych w strukturach reprogramowalnych FPGA narażonych na oddziaływanie cząstek jonizujących. W układach FPGA dominująca liczba komórek pamięci związana jest ze sterowaniem zasobami programowalnymi obejmującymi połączenia programowalne oraz realizowane funkcje logiczne. Podjęte przez doktoranta zagadnienie diagnostyki stanowi istotny obszar wyzwań dla diagnostyki układów reprogramowalnych. Specyfika uszkodzeń wywołana modyfikacją pamięci konfiguracji wymaga odpowiedniego modelowania i analizy uszkodzeń konfiguracji, co znalazło odbicie w przedstawionej pracy. Do modelowania błędów mogących wystąpić w układzie FPGA, autor opracował metodę modyfikacji opisu strukturalnego danego w języku VHDL. Pozwoliło to uzyskać precyzyjne odwzorowanie powstających błędów w strukturze układu. Przedstawione podejście modelowania uszkodzeń ma charakter nowatorski w stosunku do stosowanych w literaturze koncepcji wywodzących się z metod testowania układów ASIC. Celem rozprawy było również opracowanie metod diagnostyki dla procesorów implementowanych w strukturach FPGA wykorzystujących program diagnostyczny. Istotą opracowanej metody testowania jest ujawnienie błędu poprzez dobór odpowiedniego ciągu instrukcji oraz argumentów. W celu skrócenia procesu diagnostycznego autor przedstawia heurystyczne algorytmy minimalizacji zbioru testowego.

Uwzględniając powyższe stwierdzam że, praca doktorska mgr. inż. Mariusza Węgrzyna wpisuje się w aktualny nurt diagnostyki systemów cyfrowych implementowanych w układach reprogramowalnych FPGA narażonych na działanie promieniowania jonizującego.

2. Zawartość i ocena merytoryczna

Recenzowana rozprawa doktorska przedstawia sposób testowania wybranych rdzeni procesorów implementowanych w strukturach programowalnych FPGA. Autor przedstawił oryginalne podejście do modelowania błędów w strukturach programowalnych powstałych w wyniku oddziaływania cząstek jonizujących oraz konstruowania i optymalizacji programów diagnostycznych. Dokonał również wyczerpującej analizy maskowania błędów w opracowanych testach i sposobów ich wykrywania

Rozprawa stanowi spójną tematycznie całość i składa się z 11 rozdziałów wraz ze wstępem (rozdział 1), podsumowaniem (rozdział 10) i dodatkami (rozdział 11). Rozdziały od

2 do 4 stanowią wprowadzenie, natomiast właściwe przedstawienie uzyskanych wyników prac badawczych i implementacyjnych znajduje się w rozdziałach od 5 do 9.

W rozdziale pierwszym znajdujemy szerokie wprowadzenie w zagadnienie. Na wstępie autor kreśli zarys problemu oraz motywację jego podjęcia. Pozwoliło mu to na sformułowanie dwóch tez, które były przedmiotem badań a ich efekty zostały przedstawione w dalszej części pracy.

Rozdział drugi zawiera opis błędów wywoływanych w cyfrowym układzie elektronicznym przez promieniowanie jonizujące. Przedstawiono tam mechanizm generacji nośników w wyniku zderzenia cząstek jonizujących ze strukturą półprzewodnikową układu. W wyniku zderzenia w obszarze elementu aktywnego może dojść do wywołania krótkich impulsów pasożytniczych. W przypadku układów programowalnych powstałe impulsy pasożytnicze mogą prowadzić do zmiany stanu komórki pamięci konfiguracyjnej, a w konsekwencji modyfikacji zaprogramowanej struktury układu i powstania błędów w działaniu. W celu efektywnej analizy wywołanych uszkodzeń konieczne jest opracowanie odpowiedniego modelu. Autor analizuje modele przedstawione w literaturze. Bazują one na sklejeniu z 0 lub 1 odpowiednich węzłów układu. Autor wykazał, że metoda bazująca na sklejeniu z wartością logiczną uniemożliwia pełne odwzorowanie wszystkich możliwych błędów wywołanych zmianą stanu komórki pamięci konfiguracyjnej w blokach tablicowych LUT. Stanowi to istotny i oryginalny wkład autora.

Rozdział trzeci przedstawia zagadnienie testowania procesorów. Metody klasyczne polegające na wbudowaniu układu testującego w strukturę procesora pozwalają na przeprowadzenie testowania poszczególnych bloków funkcjonalnych. Zasady konstruowania układów generacji pobudzeń testowych oraz kompaktacji odpowiedzi są dobrze ugruntowane. W przypadku procesorów osadzonych w strukturach programowalnych testowanie strukturalne nie jest możliwe. Testowanie odbywa się za pomocą specjalnie przygotowanego programu. Autor wyróżnia podejście strukturalne oraz funkcjonalne. Istotą opracowania programu diagnostycznego jest dobór instrukcji i operandów pozwalający na pobudzenie możliwie największej liczby elementów procesora w taki sposób aby ich niepoprawne działanie zostało ujawnione w uzyskanym wyniku. Autor wskazuje na bardzo interesujący system wbudowanej diagnostyki programowej MIHST. Przedstawiona koncepcja jest ukierunkowana na ograniczenie zasobów programowalnych koniecznych do wbudowania generatora instrukcji testowych oraz kompaktora odpowiedzi uzyskanych w wyniku ich wykonania.

Zagadnienie związane z modelowaniem struktury mikroprocesora z kontrolowanym wprowadzaniem uszkodzeń na potrzeby oceny skuteczności procesu testowania została przedstawiona w rozdziale 4. Powszechne wykorzystanie języków opisu sprzętu ułatwia wszechstronne modelowanie układów cyfrowych. Najbardziej wskazanym opisem na potrzeby diagnostyki jest opis strukturalny. Autor przedstawia również metody wprowadzania

uszkodzeń bezpośrednio do strumienia konfigurującego układ programowalny. Ciekawym elementem uzupełniającym jest wykorzystanie rekonfiguracji częściowej układu na potrzeby zamodelowania uszkodzenia. Ze względu na brak otwartej dokumentacji opisującej poszczególne elementy strumienia konfiguracji opracowanie metod jej modyfikacji jest trudne i może prowadzić w skrajnych przypadkach do uszkodzenia układu.

Dalsza część pracy przedstawia opracowane przez autora metody diagnostyki procesorów wbudowanych. Jej podstawą jest zaproponowana w rozdziale 5 metoda pobudzenia ścieżek danych. Pobudzenie następuje poprzez dobranie instrukcji oraz odpowiednich argumentów. Autor również przyjął, że konstruowany test powinien posiadać własności bijektywne. Test jest konstruowany w taki sposób aby uzyskać jednoznaczne przypisanie wyniku do argumentu testu. Metody uzyskania testu bijektywnego zostały opracowane i opisane przez autora a także opatrzone wieloma przykładami. Jednoznaczne powiązanie pomiędzy wektorem testowym a wektorem odpowiedzi na test pozwala na łączenie bloków testowych w dłuższy bloki wykorzystując cykliczne przekazywanie danych pomiędzy poszczególnymi blokami. Ostatecznie zaproponowany test składa się ze zbioru instrukcji oraz z wyczerpującego zbioru argumentów. W przypadku mikroprocesora o 8-bitowych argumentach podejście wyczerpujące jest akceptowalne, jednakże w przypadku mikroprocesorów o słowie dłuższym, np. 32- lub 64-bitowym jest już nie do przyjęcia ze względu na czas konieczny do wykonania programu diagnostycznego.

Odpowiedź na pojawiające się wątpliwości co do czasu trwania testu w jednostkach o dłuższym słowie danych pojawia się w rozdziale 6. W rozdziale tym autor podejmuje udaną próbę zredukowania liczby wektorów testowych zapewniającej pokrycie wykrywanych błędów. W celu dokonania klasyfikacji oraz wyboru testów, tworzone jest pokrycie błędów przez poszczególne testy. Zaproponowano trzy heurystyczne algorytmy redukcji zbioru wektorów testowych. Pierwszy z algorytmów wykorzystuje podejście zachłanne. Wektory o największym pokryciu błędów wybierane są w pierwszej kolejności. Podejście drugie jest biegunowo odmienne. Wynikowy zbiór testowy budowany jest począwszy od wektorów pokrywających pojedynczy błąd. Ostatecznie trzeci algorytm łączy w sobie cechy poprzedników uzupełniając proces wyboru o częściową strategię „dziel i rządź”. Rozważany przez autora problem przywodzi problem minimalnego pokrycia funkcji logicznych. Przedstawione przez autora metody minimalizacji pozwoliły na 9-krotną redukcję zbioru wektorów testowych. Należy oczekiwać, że w przypadku jednostek o słowie dłuższym redukcja zbioru testowego będzie znacznie większa, co pozwoli na wykonanie testu w akceptowalnym czasie. Tytuł rozdziału mówiący o optymalnej redukcji liczby testów (Optimal reduction of number of test vectors) jest niezręcznie sformułowany. Zgodnie z tytułem, autor powinien przedstawić dowód optymalnego wyboru wektorów testowych np. wykorzystując algorytm wyznaczania minimalnego pokrycia funkcji logicznej. W rzeczywistości zostały przedstawione

algorytmy heurystyczne umożliwiające redukcję liczby testów, jednakże optymalność uzyskanego wyniku nie została określona.

W rozdziale 7 autor dokonuje oceny opracowanych metod w zastosowaniu do mikroprocesora 32-bitowego microBlaze. Opracowane do tej pory rozwiązania były weryfikowane z wykorzystaniem mikroprocesora picoBlaze, którego struktura wewnętrzna jest znana. Pozwoliło to na modelowanie błędów występujących w strukturze, obserwacje ich wpływu na działanie mikroprocesora oraz uzyskane wyniki testów. W przypadku mikroprocesora microBlaze struktura jest chroniona przez dostawcę (Xilinx) i nie jest dostępna w sposób otwarty. Modelowanie oraz wykrycie błędów jest możliwe jedynie metodami programistycznymi gdzie fragment programu odpowiada za modelowanie uszkodzenia danych. Nie jest możliwa ocena pokrycia błędów występujących w strukturze układu. Dodatkowo, przeprowadzenie testowania mikroprocesora 32-bitowego ujawniło znaczne nakłady czasu konieczne na wykonanie wyczerpujących programów testowych (konieczność wyliczenia wyników dla 2^{32} wektorów wejściowych). Autor uznaje, że opracowane metody dla mikroprocesora picoBlaze można wykorzystać do innych mikroprocesorów implementowanych w układach FPGA. Postawiona przez autora teza ogólności wymaga szerszego udowodnienia. W szczególności w kontekście zastosowania złożonych mechanizmów adresowania, warunkowego wykonania instrukcji czy też arytmetyki zmiennoprzecinkowej. Wybór mikroprocesora microBlaze ogranicza obiektywną ocenę metod testowania ze względu na brak dostępu do struktury. Wydaje się, że w tym przypadku autor mógł podjąć próbę przeanalizowania podobnego rdzenia udostępnianego w otwartym opisie funkcjonalnym lub strukturalnym.

Z punktu widzenia oceny efektywności testu bardzo istotna jest zdolność modelowania uszkodzeń w strukturze układu. W rozdziale 8 przedstawiono metodę modelowania uszkodzeń w blokach tablicowych LUT. Modelowanie wykorzystuje modyfikację parametru generycznego określającego zawartość tablicy bloku LUT. Autor zaproponował sterowną skryptami w języku Perl generację uszkodzonej struktury mikroprocesora. W pracy zabrakło jednak odniesienia się do struktury plastra konfigurowalnego bloku logicznego CLB. Stanowi on podstawową komórkę skupiającą bloki tablicowe, przerzutniki oraz blok szybkich przeniesień arytmetycznych.

Zaproponowane metody diagnostyczne nie pozwalają osiągnąć 100% pokrycia uszkodzeń. Autor dokonuje analizy pokrycia diagnostycznego błędów w rozdziale 9 pracy. Na podstawie przeprowadzonych badań mikroprocesora picoBlaze dokonuje analizy szczegółowej maskowania uszkodzeń w strukturze logicznej mikroprocesora. Pozwala to określić szczegółowe warunki pobudzenia bloku tablicowego w celu wyprowadzenia błędnej pozycji pamięci konfiguracyjnej. Przedstawione rozważania pozwalają na uzupełnienie zbioru testowego o elementy zwiększające wykrywalność uszkodzeń.

Podsumowanie efektów prac autor dokonuje w rozdziale 10. **Na podstawie lektury pracy można stwierdzić, że postawione w rozdziale 1 tezy zostały udowodnione przez autora.** W szczególności utworzenie sekwencji testowej o wzajemnie jednoznacznym przyporządkowaniu argumentu i wyniku. Autor wykazał również skutecznie prawdziwość drugiej z tez, przedstawiając metody minimalizacji zbioru wektorów testowych.

Wykaz literatur zawierający 92 pozycje należy uznać za wyczerpujący. W zestawieniu brak pozycji z ostatnich lat. Najnowsze publikacje dotyczące tematyki testowania systemów cyfrowych implementowanych w reprogramowalnych układach logicznych pochodzą z roku 2017. W spisie literatury odnotowano brak dokumentów przedstawiających architektury układów programowalnych wybranych rodzin.

3. Uwagi krytyczne, wątpliwości, pytania

Podczas obrony publicznej chciałbym poznać zdanie doktoranta na poniższe wątpliwości:

1. W mikroprocesorze możemy wyróżnić ścieżkę sterowania oraz ścieżkę danych. W przedstawionej pracy, autor nie odniósł się do sposobu diagnostyki jednostki w przypadku uszkodzeń w ścieżce sterowania. Uszkodzenie takie prowadzi do zaburzenia procesu wykonywania instrukcji. Przyjęta koncepcja blokowego przetwarzania danych wymaga poprawnego działania ścieżki sterowania w celu wykonania testu. Jaki będzie miało wpływ na wykonanie programu testowego uszkodzenie w obszarze ścieżki sterowania prowadzące do zaburzenia pobierania instrukcji i w jaki sposób zostanie rozpoznane.
2. Opracowana metoda diagnostyki wymaga uruchomienia opracowanego programu testowego. Jak pogodzić kwestię umieszczenia programu diagnostycznego a silnie ograniczoną przestrzenią pamięci programu, jak ma to miejsce w przypadku mikroprocesora picoBlaze
3. Przedstawione przez autora metody optymalizacji zbioru testowego znakomicie sprawdzają się w przypadku mikroprocesora o słowie 8-bitowym. Jakie skuteczne obliczeniowo podejście należy zastosować w przypadku zbiorów testowych dla mikroprocesorów o słowach 32-bitowych.
4. Przedstawiona metoda pobudzenia ścieżki danych jest bardzo ciekawa. Czy autor rozważał sformalizowanie metody bazującej na połączeniu metody pobudzenia uszkodzenia, z której wywiedziona zostaje instrukcja oraz argumenty operacji. Wniosek taki nasuwa się czytelnikowi po zapoznaniu się ze szczegółową analizą niewykrytych błędów w rozdziale 9 pracy

4. Uwagi szczegółowe

Przedstawiona rozprawa ma również swoje słabe strony i pewne niedociągnięcia. Chciałbym podkreślić, że znaczna część uwag ma charakter polemiczny.

Praca napisana została w stronie czynnej np. „I followed” czy „I handle this situation programmatically”. Przyjęta przez autora forma odbiega od zwyczajowo przyjętej dla prac naukowych, w których stosuje się stronę bierną.

Równania na stronie 23 (30, 149, 150,...) zostały zapisane jako akapit tekstu. Równania powinny zostać przedstawione w ogólnie przyjętym zapisie równań matematycznych i utworzone za pomocą odpowiedniego edytora. Zmienne powinny zostać opisane za pomocą oznaczeń przyjętych przez autora i wyjaśnione podczas pierwszego ich użycia.

Rysunek 2.2.8 ma przedstawiać rejestr (komórkę pamięci) z mechanizmem zatraskowym (latch). W schemacie tym nie uwzględniono kluczy tranzystorowych umożliwiających przełączanie układu pomiędzy stanem pamięci oraz wprowadzenia danych.

Rysunek 2.2.9 przedstawia rejestr zawierający model generacji ładunku przez cząsteczkę jonizującą. Model używa siły elektromotorycznej ($R = 0$), co uniemożliwia zamodelowanie impulsu wywołanego poprzez wstrzyknięcie ładunku do kondensatora. Model powinien zostać wywiedziony z tranzystora MOSFET ze sterowanymi siłami prądomotorycznymi, co umożliwi poprawne zamodelowanie rozkładu napięć.

W rozdziale 2.2.2 rozważano propagację sygnału błędnego w wyniku generacji impulsu przez cząstkę jonizującą. Przedstawiony schemat układu logicznego składa się z bramki sumy oraz iloczynu. Analizowana struktura nie odpowiada fizycznej implementacji układowej. W tym przypadku autor powinien wykorzystać elementy podstawowe technologii CMOS czyli bramki NAND lub NOR. W przypadku przedstawionego układu można sugerować wykorzystanie bramek NAND.

Podpisy pod rysunkami, tabelami: 2.2.6, 2.2.7, 2.2.9, 2.3.1, 2.3.2, 2.3.4, 5.4.1 rozpoczynają się od nazwy rozdziału i nie są adekwatne dla podpisu i treści rysunku lub tabeli

Brak odsyłaczy do opisu architektury układu FPGA w nocie katalogowej producenta np. str. 283: „For example, the Xilinx Virtex 5+ TM”

Brak odsyłacza do Architecture Description Language (str. 42₁₀).

Rysunek 4.3.2 zawiera bramkowanie sygnału zegarowego – w przypadku implementacji w układzie FPGA należy unikać takiego rozwiązania zastępując go sygnałem CE (Clock Enable) przerzutnika.

Autor wielokrotnie odwołuje się do elementów przedstawionych w dalszej części pracy np. str597: „Result Analyser consist of script programs written in C according to principles described in chapter 7.”. Utrudnia to lekturę pracy.

Str. 65⁷: “In the test sequence, each processor instruction participates at least once (in order e.g. to test the instruction decoder of the processor core).” – W procesorach instrukcja umożliwia wykorzystanie wielu trybów adresowania. Pojęcie instrukcji w tym przypadku jest niejednoznaczne.

str. 68₅, 107₆: źródłem definicji matematycznych nie powinna być Wikipedia

str. 111 rysunek 6.8 – „Bijective cyclic block to the SL1 testing” – podpis nieadekwatny do treści. Na rysunku widnieje rejestr LFSR. Należy zwrócić uwagę, że przedstawiony układ posiada $2^n - 1$ stanów pod warunkiem, że stan początkowy jest różny od 0

str. 123₄: “MicroBlaze can be clocked from 100 MHz to 700 MHz” – brak wskazania dla jakich warunków/układów FPGA oraz na podstawie jakiego dokumentu możliwa jest do osiągnięcia częstotliwość taktowania rdzenia mikroprocesora wynosząca 700MHz.

str. 146₃ „FCmax. Other resources as i.e. switches, programmable interconnections are not available from Xilinx programming tools level or even access is protected against people outside the corporation.” – struktura wewnątrz jest dostępna za pomocą języka XDL (Xilinx Design Language) dostępnego w pakietach oprogramowania ISE.

str. 149¹: równanie logiczne w zapisie tekstowym. Autor niekonsekwentnie zapisuje negację uprzednio za pomocą tyldy natomiast w tym przypadku za pomocą wykrzyknika. Podobna niekonsekwencja w równaniach na str150

str. 177¹⁴: brak referencji do architektury układów PFGA, kart katalogowych rdzeni microBlaze i picoBlaze.

4.1. Nieakceptowalne sformułowania

- str. 10¹⁵: „wymaga bijektywnej relacji „jeden do jednego”” – typowy pleonazm
- str. 23²: “at the end of OR gate” – the OR gate output
- str. 27₆: FPGA consists of an $N \times N$ array of configurable logic blocs (CLB) – powinno zostać przedstawione podejście ogólne $R \times C$ (Row x Column)
- str 40₁: “top layer of the chip” – top level of the chip
- str. 43₆: “using the so-called Specman Elite block” – istotą jest wykorzystanie język “e” opracowanego przez firmę Verisity natomiast Specman Elite to narzędzie diagnostyczne wykorzystujące język „e” oraz modele HDL.
- str 149₄:”by additional I2 ingredient” – by additional element I2 (of the logic sum)

- str 54₁₀: Hardware-based techniques utilize a *VHDL* models implemented in FPGA – zamiast VHDL powinno zostać użyte ogólne sformułowanie HDL
- str. 109 rysunek 6.6: Circuit representation of 8-bit Fibonacci LFSR – na rysunku przedstawiono rejestr LFSR (brak związku z ciągiem Fibonacciego), na schemacie brak zwyczajowego oznaczenia wejść i wyjść przerzutników.
- Str. 155₁ – there appears 0x10 – zwrot żargonowy można zastąpić the result is equal 0x10
- Str. 160₄ – “can open concurrently Xilinx FDR flips flops” – co oznacza pojęcie otwarcia współbieżnego rejestrów
- str. 164₂: There is no way to get a wrong result – It is not possible

4.2. Niedociągnięcia edytorskie, drobne błędy

- str. 12¹⁴ (Xilinx 2019) nieistniejący odsyłacz
- str 14¹ then powinno być ten
- str 22⁵ flip flop powinno być flip-flop
- Równania (np. str. 23) nie zostały ponumerowane co utrudnia odwołanie się do odpowiedniego równania
- str. 26 podpis pod rysunkiem jest: effectin a SRAM powinno być: effect in a SRAM
- str. 28 jest 2n powinno być 2ⁿ – n powinno być wykładnikiem
- str. 30 – funkcje logiczne zostały zapisana za pomocą symboli tekstowych podczas gdy powinna zostać przedstawiona w zapisie algebraicznym (np. utworzone za pomocą edytora równań). Autor powinien także przestrzegać zasad oznaczenia zmiennych, które zwyczajowo zapisuje się kursywą i małymi literami. Litery duże zarezerwowane są dla sygnałów co powinno znaleźć odzwierciedlenie w zapisie oraz odpowiednim komentarzu o skojarzeniu sygnałów logicznych i zmiennych dwustanowych.
- str 39₁₇: “2 to the power of 32” – 2³²
- str. 45 rysunek 3.2.6 – brak krawędzi prowadzącej do wierzchołka E
- str. 63 rysunek 4.3.6 – na schemacie widoczna bramka OR podpisana jako XOR
- str.72₁₁: of „CARRY” flags – CARRY flag
- str. 107₇: event – even
- str. 109₈: „X8 + X6 + X5 + X4 + 1” powinno być (wykładniki jako indeks górny): X⁸ + X⁶ + X⁵ + X⁴ + 1
- str. 118₁: „According the MicroBlaze user guide” – brak odsyłacza do noty katalogowej rdzenia np. UG081, Xilinx, 2010.
- str. 162₃: waives powinno być waves
- str 164₁: 6th powinno być 6th
- str. 171¹: then powinno być ten

Chciałbym podkreślić, że powyższe uwagi dotyczące użytego słownictwa oraz drobnych braków edytorskich, wyspecyfikowane z opiniodawczego obowiązku, absolutnie nie podważają zasadniczego dorobku Autora i nie obniżają mojej wysokiej oceny wartości merytorycznej rozprawy.

5. Ocena końcowa rozprawy

Uważam, że przedstawione w rozprawie doktorskiej oryginalne pomysły oraz wyniki eksperymentów wnoszą istotny wkład w rozwój dyscypliny Automatyka, Elektronika i Elektrotechnika. Oprócz przedstawionej rozprawy doktorant jest autorem 6 publikacji naukowych prezentujących zagadnienia dotyczące diagnostyki systemów cyfrowych implementowanych w strukturach programowalnych.

Podsumowując stwierdzam, że pomimo pewnych niedociągnięć, rozprawa doktorska Pana mgr. inż. Mariusza Węgrzyna spełnia wymagania stawiane w ustawie "Prawo o szkolnictwie wyższym i nauce", zatem **wnoszę o dopuszczenie rozprawy doktorskiej do publicznej obrony i dalszych etapów przewodu doktorskiego w dyscyplinie Automatyka, Elektronika i Elektrotechnika.**

Adam Miłik