

dr hab. inż. Dariusz Makowski, profesor uczelni
Katedra Mikroelektroniki i Technik Informatycznych
Wydziału Elektrotechniki, Elektroniki, Informatyki i Automatyki
Politechnika Łódzka
ul. Wólczńska 221, 93-005 Łódź

Łódź, dn. 13.11.2023 r.

RECENZJA ROZPRAWY DOKTORSKIEJ

Tytuł rozprawy: „Functional Bus Description Language”.

„Język Opisu Funkcjonalnych Magistral”.

Autor rozprawy: mgr inż. Michał Kruszewski

1. Podstawa prawna, przedmiot i zakres recenzji

Niniejsza recenzja rozprawy doktorskiej została przygotowana na podstawie pisma z dnia 27.06.2023 r. sformułowanego na podstawie uchwały Rady Naukowej Dyscypliny Automatyka, Elektronika, Elektrotechnika i Technologie Kosmiczne Politechniki Warszawskiej.

Przedmiotem recenzji jest rozprawa doktorska mgr inż. Michała Kruszewskiego „Functional Bus Description Language”. Promotorem rozprawy jest dr hab. inż. Wojciech Zabołotny, prof. uczelni.

Recenzja została opracowana w oparciu o Ustawę oraz Rozporządzenia Ministra Nauki i Szkolnictwa Wyższego:

- Rozporządzenie Ministra Nauki i Szkolnictwa Wyższego z dnia 19 stycznia 2018 r. w sprawie szczegółowego trybu i warunków przeprowadzania czynności w przewodzie doktorskim, w postępowaniu habilitacyjnym oraz w postępowaniu o nadanie tytułu profesora (Dz. U. z 2018 r., poz. 261).
- Ustawa z dnia 20 lipca 2018 roku - Prawo o szkolnictwie wyższym i nauce (Dz. U. z 2018 r., poz. 1668 z późniejszymi zmianami).
- Rozporządzenie Ministra Nauki i Szkolnictwa Wyższego z dnia 20 września 2018 r. w sprawie dziedzin nauki i dyscyplin naukowych oraz dyscyplin artystycznych (Dz. U. z 2018 r., poz. 1818 z późniejszymi zmianami).

2. Jakie zagadnienie naukowe jest rozpatrzone w pracy /teza rozprawy/ i czy zostało ono dostatecznie jasno sformułowane przez autora? Jaki charakter ma rozprawa (teoretyczny, doświadczalny, inny)?

Rozprawa doktorska dotyczy ważnej i aktualnej tematyki projektowania złożonych systemów cyfrowych wykorzystujących zarówno systemy mikroprocesorowe, jak również układy programowalne FPGA (Field Programmable Gate Array). Dynamiczny postęp

mikroelektroniki pozwolił na integrację procesora (lub rdzeni kilku procesorów) oraz złożonej struktury programowalnej w tym samym układzie scalonym, określanym często terminem System On Chip (SoC). Takie podejście pozwala na wykorzystanie klasycznych rozwiązań bazujących na komputerach oraz złożonych układów cyfrowych lub układów ASIC dostarczających specyficzną funkcjonalność, której nie jest w stanie zapewnić procesor. Systemy wykorzystujące procesory oraz układy programowalne stają się coraz bardziej popularne ze względu na ich duże możliwości obliczeniowe oraz coraz niższe koszty.

W takich systemach, projektant równolegle rozwija oprogramowanie struktury logicznej-układu FPGA (gateway) jak również oprogramowanie wysokiego poziomu, które realizowane jest z użyciem procesora (firmware, software). Jednym z ważniejszych aspektów podczas budowania systemów jest zaprojektowanie oraz dalsze rozwijanie (utrzymanie) interfejsu pomiędzy procesorem, a układem programowalnym. Układy zwykle połączone są przy pomocy wydajnej magistrali systemowej. W takim przypadku podsystemy układu programowalnego mogą zostać bezpośrednio zamapowane na przestrzeń pamięci procesora, podobnie jak pamięci oraz inne urządzenia peryferyjne procesora. Komunikacja odbywa się z wykorzystaniem pliku rejestrów (rejstry wraz z przypisanymi adresami) zaimplementowanych w układzie programowalnym.

Autor uczestniczył podczas projektowania, implementacji oraz utrzymania oprogramowania niskiego oraz wysokiego poziomu dedykowanego do akwizycji danych w eksperymencie CBM realizowanym w instytucie Badań Ciężkich Jonów GSI w Darmstadt, Niemcy. Proces rozwoju oprogramowania trwał kilkanaście lat i wymagał ciągłej aktualizacji oraz modyfikacji interfejsu pomiędzy specjalizowanymi układami, a procesorem zrealizowanego w postaci bloku rejestrów. Podczas realizacji tego projektu Autor starał się maksymalnie uprościć oraz zautomatyzować proces implementacji interfejsu pomiędzy układami oraz generacji oprogramowania niskiego i wysokiego poziomu. Pomimo użycia programu AGWB (Address Generator for WishBone) do automatycznej generacji rejestrów wciąż zachodziła potrzeba ręcznego opracowania oraz utrzymania oprogramowania do komunikacji, co było procesem żmudnym oraz podatnym na błędy. Implementacja nowego rejestru wymagała aktualizacji całej mapy adresów zarówno po stronie układu FPGA, jak również po stronie procesora.

Autor zaproponował nowe podejście pozwalające na funkcjonalne opisanie rejestrów przy pomocy języka programowania FBDL (Functional Bus Description Language) oraz wykorzystanie kompilatora do automatycznej generacji rejestrów oraz oprogramowania do ich obsługi. Głównym celem rozprawy było opracowanie języka umożliwiającego opis magistrali systemowej i rejestrów poprzez określenie funkcjonalności danych. Praca obejmuje również implementację prototypu kompilatora pozwalającego na weryfikację koncepcji, a także omówienie niektórych ogólnych szczegółów implementacji, z którymi prawdopodobnie będzie musiał się zmierzyć kompilator zgodny z językiem FBDL.

Zagadnienie naukowe omawiane w pracy zostało sformułowane w postaci jednej tezy zamieszczonej w rozdziale 4.1 na stronie 54 (tłumaczenie polskie oraz wersja oryginalna w jęz. angielskim):

„Strukturę magistrali i rejestru można wywnioskować na podstawie opisu funkcjonalnego danych, które mają być przechowywane w rejestrach. Co więcej, takie podejście oferuje pewne istotne zalety w większości typowych przypadków użycia w porównaniu z klasycznym podejściem, w którym jawnie opisana jest struktura rejestru”.

Thesis:

„It is possible to infer the bus and register structure based on the description of the functionality of the data that shall be stored in the registers. Moreover, such an approach offers some significant advantages in most typical use cases compared to the classic approach in which register structure is described explicitly”.

Zarówno założenia jak i cel pracy zostały jasno sformułowane przez Autora. Rozprawa ma charakter praktyczny. Autor zaproponował nowe podejście bazujące na opisie funkcjonalnym. W ramach przeprowadzonych badań naukowych opracował język FBDL pozwalający opisać magistralę systemową oraz rejestry, jak również prototyp kompilatora pozwalający na zbadanie przydatności zaproponowanej metodyki projektowania układów cyfrowych. W pracy autor przedstawił teoretyczne rozważania oraz wnioski podsumowujące zalety jak również ograniczenia rozwiązania opartego na funkcjonalnym opisie.

3. Czy w rozprawie przeprowadzono w sposób właściwy analizę źródeł /w tym literatury światowej, stanu wiedzy i zastosowań w przemyśle/ świadczącej o dostatecznej wiedzy autora? Czy wnioski z przeglądu źródeł sformułowano w sposób jasny i przekonujący?

Rozprawa doktorska składa się z 9 rozdziałów poprzedzonych wykazem skrótów, krótkim wprowadzeniem, wykazem literatury oraz czterema załącznikami. W załączniku 4 Autor zamieścił opis opracowanego języka FBDL (44 strony).

Układ merytoryczny pracy obejmuje część teoretyczną oraz praktyczną składającą się z opisu specyfikacji funkcjonalnej oraz implementacji kompilatora. Wprowadzenie zawiera, krótki opis metodyki projektowania oraz implementacji systemów cyfrowych zbudowanych z wykorzystaniem układów programowalnych (FPGA) współpracujących z systemem mikroprocesorowym realizującym program, właściwy kontekst oraz uzasadnienie podjęcia badań naukowych przez Autora w tej tematyce.

W rozdziale 1, stanowiącym wstęp, Autor opisuje metody łączenia funkcjonalnych bloków systemu cyfrowego przy użyciu magistral systemowych oraz przedstawia kilka przykładów rozwiązań wykorzystywanych podczas programowania układów FPGA oraz układów SoC. Autor przedstawia metody projektowania, implementacji rejestrów oraz wybrane aspekty ich modyfikacji. W rozdziale przedstawiono porównanie interfejsów układów cyfrowych dla 2 przykładowych systemów bazujących na podejściu rejestrowym oraz, zaproponowanym przez Autora opisie funkcjonalnym, jak również omówiono podstawowe wady pierwszego rozwiązania. Autor opisuje przykład rozwiązania funkcjonalnego z wykorzystaniem zaproponowanego języka FBDL.

W rozdziale 2 Autor opisuje architekturę systemów cyfrowych oraz komputerów połączonych przy pomocy magistral, opisuje ich budowę oraz przedstawia przykłady wykorzystywanych rozwiązań. Autor definiuje pojęcie magistrali oraz omawia przykłady złożonych magistral równoległych oraz szeregowych wykorzystywanych w komputerach. W dalszej części bardzo ogólnie zostały opisane standardy wykorzystywane w mikrokontrolerach i układach SoC: AMBA AXI, Wishbone oraz NoC. Autor przedstawia przykładowe przebiegi czasowe obrazujące transmisję zapisu lub odczytu danych.

W rozdziale 3 opisano analizę aktualnego stanu wiedzy dotyczącą rozwiązań bazujących na paradygmacie rejestrowym. Autor zamieścił porównanie 16 narzędzi programowych pozwalających na zarządzanie magistralami, implementację rejestrów, bardziej złożonych

struktur danych jak również przeprowadzenie operacji matematyczno-logicznych, automatyczne generowanie szablonów funkcji do obsługi rejestrów w języku opisu sprzętu (VHDL, Verilog, SystemVerilog) oraz języków wysokiego poziomu (C, C++, Python, HTML, etc.) wraz z dokumentacją. Wśród opisanych narzędzi SystemRDL jest językiem pozwalającym na opisanie rejestrów oraz umożliwia przypisanie dodatkowego znaczenia do przechowywanych danych. Po wykonaniu przeglądu stanu wiedzy Autor stwierdza, że wszystkie wykorzystywane obecnie rozwiązania bazują na opisie rejestrowym. Brakuje w literaturze rozwiązań bazujących na podejściu funkcjonalnym.

W rozdziale 4 Autor definiuje problem badawczy oraz przedstawia pojedynczą tezę badawczą. Celem badań naukowych jest opracowanie oraz zbadanie języka umożliwiającego opis magistrali systemowej i rejestrów poprzez określenie funkcjonalności danych. Na potrzeby badań Autor opracował prototyp kompilatora pozwalającego na zbadanie zaproponowanej metodyki projektowania magistral oraz rejestrów bazując na opisie funkcjonalnym.

W rozdziale 5 Autor opisuje specyfikację funkcjonalną języka FBDL obejmująca wykorzystanie: bloków logicznych oraz ich parametryzacji, magistral systemowych, bloków konfiguracyjnych, obsługi przerwań, obsługi pól bitowych, pamięci, bloków proceduralnych jak również strumieniowych oraz ich parametryzacji, rejestrów statusowych oraz zwracanych wartości.

W rozdziale 6 Autor opisał listę metod, których język FBDL obecnie nie wspiera obejmujących: dwustronny dostęp do danych, typów enumerowanych, funkcji obliczeniowych oraz atrybutów definiowanych przez programistę, jak również ręcznego przydzielania adresów. Autor zamieścił również uzasadnienie dlaczego te metody oraz funkcje nie są wspierane.

W rozdziale 7 Autor opisał budowę oraz wybrane szczegóły dotyczące implementacji prototypowego kompilatora języka FBDL opracowanego na potrzeby badań przeprowadzonych w ramach pracy doktorskiej. W pierwszej części autor opisał front-end kompilatora oraz kolejne fazy kompilacji oraz budowy rejestrów, algorytm wykorzystany do ich implementacji. W drugiej części zamieścił opis dotyczący back-end'u kompilatora dotyczący generowania programów obsługujących funkcje zapisu, odczytu rejestrów oraz odczyt-modyfikacja-zapis.

W rozdziale 8 „Praktyczne użycie” Autor zamieścił wyłącznie informację dotyczącą wykorzystania kompilatora FBDL podczas opracowania modułu generatora opóźnień dla femtosekundowego lasera realizowanego w ramach projektu Eurostars-2 „Opracowanie układu optycznego do szybkiej laserowej obróbki materiałów przezroczystych” przez firmę Fluence Sp. z o.o. W rozdziale zawarto wyłącznie informację o tajności prac prowadzonych w ramach ww. projektu. Rozdział nie wnosi żadnej wartości merytorycznej, ani dowodów wspierających postawioną tezę.

W rozdziale 9 Autor podsumowuje zalety zaproponowanego podejścia funkcjonalnego oraz języka FBDL w odniesieniu do klasycznego podejścia bazującego na rejestrach.

Autor odwołuje się w pracy do 84 pozycji literaturowych. W pracy występuje niewielka liczba odwołań do źródeł naukowych wnoszących wartość naukową do pracy (5 pozycji książkowych, 24 artykuły naukowe). Około połowy stanowią artykuły pochodzące z czasopism z listy filadelfijskiej. Natomiast znaczną większość (ponad 65%) stanowią odwołania do źródeł elektronicznych udostępnionych przez firmy oraz ośrodki naukowo-badawcze obejmujących: repozytoria Git (19 pozycji), 2 raporty, 1 prezentacja, 9 standardów

oraz 24 pozycje obejmujące artykuły popularno-informacyjne, instrukcje obsługi, materiały szkoleniowe, itd.

W pracy przeprowadzono analizę stanu wiedzy. W rozdziale drugim Autor opisał magistrale wykorzystywane w systemach komputerowych do łączenia bloków systemu cyfrowego, natomiast w kolejnym rozdziale zamieścił porównanie 16 narzędzi oraz metod pozwalających na uproszczenie procesu implementacji oraz utrzymania rejestrów. Dokonana analiza dotyczy głównie funkcji wspieranych przez analizowane narzędzia oraz dostępności formalnej specyfikacji.

Pewien niedosyt pozostawia fakt, że w znacznej części (65%) są do odwołanie do źródeł o niskiej wartości naukowej. Jednak ze względu na praktyczny charakter pracy taka analiza stanu wiedzy jest akceptowalna.

Wnioski wynikające z przeglądu literatury są bardzo skromne i sprowadzają się do funkcjonalnego porównania narzędzi oraz stwierdzenia, że wszystkie analizowane przez Autora narzędzia operują bezpośrednio na rejestrach. Żadne z narzędzi ani metod nie pozwala na zarządzanie rejestrami mając na uwadze ich funkcjonalność.

W przeprowadzonej analizie brakuje przeglądu literatury pod kątem innej metodyki pozwalającej na opracowanie interfejsu pomiędzy układem cyfrowym, a procesorem. Autor ograniczył analizę do narzędzi, czy też bibliotek wykorzystujących rejestry. Autor nie wspomina również o metodach pozwalających na implementację rejestrów w postaci struktury, unii (język C), czy też obiektów (język C++, Python).

Wśród cytowanych prac znajdują się tylko 2 pozycje współautorstwa Doktoranta: Sensors/MDPI (IF=3,9) oraz Photonics Applications in Astronomy, Communications, Industry and High-Energy Physics Experiments/International Society for Optics and Photonics. Autor zamieścił również odwołanie do 3 repozytoriów na których umieścił fragmenty kompilatora. Opis obu źródeł jest również bardzo skromny.

4. Czy autor rozwiązał przedstawione zagadnienia, czy użył właściwej do tego metody i czy przyjęte założenia są uzasadnione?

Cel pracy (rozdział 4.2) dotyczy opracowania nowego języka pozwalającego na opisanie magistrali oraz rejestrów poprzez zdefiniowanie typu przechowanych w nich danych. Praca obejmowała również opracowanie prototypu kompilatora pozwalającego na zweryfikowanie poprawności proponowanego rozwiązania. W pracy Autor nie przedstawił metodyki badawczej jaką posługiwał się, aby osiągnąć założony cel.

W pierwszej części Autor przeprowadził przegląd interfejsów/magistral oraz metod pozwalających na implementację oraz zarządzanie rejestrami wykorzystywanymi w systemach cyfrowych. Na tej podstawie opracował wymagania w odniesieniu do nowego języka programowania pozwalającego na zarządzanie rejestrami.

Autor zamieścił opis funkcjonalności jaką powinien zapewniać język FBDL w rozdziale 5, ograniczenia funkcjonalności w rozdziale 6 oraz szczegóły dotyczące implementacji kompilatora w rozdziale 7, jak również przykłady ilustrujące opis oraz strukturę wybranych rejestrów, czy też bardziej złożonych obiektów generowanych przez kompilator. W tym rozdziale zamieścił również wnioski, spostrzeżenia dotyczące prac nad kompilatorem.

W pracy Autor skupił się głównie na opracowaniu semantyki języka FBDL oraz opisu samego kompilatora. Natomiast w pracy zabrakło spójnych przykładów ilustrujących cały proces implementacji rejestrów, w szczególności informacji dotyczących logiki

implementowanej w układzie programowalnym, przypisania adresów do rejestrów, połączenia bloku rejestrów z magistralą systemową oraz implementacji bardziej złożonych obiektów i mechanizmów opisanych w rozdziale 5.

Brakuje dobrych przykładów pokazujących całą ścieżkę implementacji rejestrów od momentu napisania programu w języku FBDL, kompilacji oraz generacji plików opisujących zaimplementowane rejestry w układzie programowalnym w języku opisu sprzętu (VHDL lub Verilog) oraz funkcji w języku wysokiego poziomu. Brakuje również podobnych przykładów obrazujących proces implementacji rejestrów z użyciem klasycznych rozwiązań bazujących na podejściu zorientowanym na rejestrach, co pozwoliłoby na porównanie obu metod oraz lepsze pokazanie zalet proponowanego rozwiązania.

W pracy brakuje również podejścia naukowego dowodzącego słuszności postawionej hipotezy badawczej. Autor powinien zastanowić się nad wykorzystaniem metod naukowych: matematycznych, analitycznych, eksperymentalnych w celu udowodnienia postawionej tezy. W pracy zawarto luźne przemyślenia oraz wnioski mające na celu udowodnienie tezy. Natomiast bardzo często są to oczywiste wnioski (*Shorter development time, as more code can be automatically generated*) lub są to dyskusyjne stwierdzenia nieoparte żadnymi dowodami (rozdział 9).

W pracy należy zamieścić twarde dowody wraz z dokładnym odniesieniem do konkretnego fragmentu uzasadnianej tezy.

Należy podkreślić, że postawiona teza obejmuje dwa problemy naukowe:

- 1) *Strukturę magistrali można wywnioskować na podstawie opisu funkcjonalności danych, które mają być przechowywane w rejestrach.*
- 2) *Strukturę rejestru można wywnioskować na podstawie opisu funkcjonalności danych, które mają być przechowywane w rejestrach.*

W pracy brakuje dowodów potwierdzających słuszność pierwszej części tezy. Wydaje się, że zaproponowany przez Autora język opisu magistrali pozwala na opisanie funkcjonalne rejestrów, natomiast nie pozwala na pełny opis funkcjonalny magistrali systemowej oraz poznanie jej struktury. W zaproponowanym języku brakuje odpowiednich typów pozwalających na opisanie struktury magistrali adresowej, danych oraz sterującej. Trudno wyobrazić sobie opis funkcjonalny bardziej złożonych magistral systemowych. W przedstawionych przykładach występują wyłącznie dwa słowa kluczowe „Main bus”. W jaki sposób można zatem poznać strukturę tak złożonych interfejsów jak: AMBA AXI, Wishbone, czy też PCI Express? Jak można poznać, czy też zmodyfikować szerokość magistrali adresowej, sterującej, zegarowej, kolejności przesyłanych bajtów, podtyp magistrali (np. AXI-Lite) oraz zweryfikować ich kompatybilność? Autor nie zamieścił również definicji „struktury magistrali”, o której mowa w pierwszej części tezy. W pracy nie zamieszczono przykładów pokazujących podłączenie bloku rejestrów do wygenerowanej magistrali, czy plików wynikowych (Provider) opisujących wygenerowaną logikę dla układu programowalnego. Zamieszczenie takich przykładów, dyskusji oraz wniosków być może odpowiedziałyby na powyższe pytania.

Drugi aspekt tezy został opisany w pracy dokładniej. Autor zamieścił częściowe kody oraz wyniki kompilacji, często umieszczone w różnych rozdziałach, np. cenne informacje znajdują się już w rozdziale 1.

Brakuje dobrych, przekonujących i kompletnych przykładów dowodzących słuszności tezy. W pracy w zasadzie nie zamieszczono żadnych wyników kompilacji języka FBDL (program

dla układu programowalnego oraz procesora). W obecnej sytuacji nie ma możliwości oceny, czy metodyka wykorzystana w pracy jest poprawna oraz, czy teza została w pełni udowodniona. Te informacje należy uzupełnić.

5. Na czym polega oryginalność rozprawy, co stanowi samodzielny i oryginalny dorobek autora, jaka jest pozycja rozprawy w stosunku do stanu wiedzy czy poziomu techniki reprezentowanych przez literaturę światową?

Za oryginalne osiągnięcie Autora w odniesieniu do obecnego stanu wiedzy należy zaliczyć przede wszystkim opracowanie metodyki projektowania interfejsu pomiędzy układem cyfrowym, a procesorem bazującej na funkcjonalnym opisie danych.

Autor opracował specyfikację nowego języka FBDL (składnia i semantyka) pozwalającego na opisanie rejestrów oraz dodatkowych metod przydatnych podczas projektowania układów cyfrowych bazując na typach zdefiniowanych w języku FBDL. Pełna specyfikacja języka została zamieszczona w załączniku D.

Autor opracował prototyp kompilatora umożliwiający automatyczne wygenerowanie bloku rejestrów opisanych w języku opisu sprzętu, przypisanie adresów oraz magistrali łączącej komponent rejestrów z systemem cyfrowym (provider code). Kompilator pozwala również na wygenerowanie dedykowanej biblioteki dostarczającej funkcje do komunikacji z rejestrami (zapis, odczyt, odczyt-modyfikacja-zapis), masek bitowych oraz innych specjalizowanych funkcji oraz automatów (requester code).

Autor opracował prototyp kompilator składający się z dwóch elementów: front-end oraz back-end. Natomiast na tym etapie nie jest jasne jakie typy, funkcje oraz mechanizmy zdefiniowane w języku FBDL są obecnie wspierane przez kompilator, np.: generowanie bloków logicznych oraz ich parametryzacja, jakie magistrale są obsługiwane, bloki konfiguracyjne, obsługa przerwań, obsługa pól bitowych, pamięci, bloków proceduralnych jak również strumieniowych oraz ich parametryzacji, rejestrów statusowych oraz zwracanych wartości.

Wart podkreślenia jest fakt, że doktorant zaprojektował oraz opracował skomplikowany kod kompilatora języka FBDL, który dostępny jest na otwartych repozytoriach (odnośniki znajdują się w pracy).

Doktorant jest współautorem 2 publikacji naukowych. Pierwsza z publikacji została opublikowana w czasopiśmie z listy JCR: MDPI, Sensors w 2021 roku. Udział Autora oceniony został na 15%. Druga praca została opublikowana w czasopiśmie: Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments (2019).

W pracy doktorskiej nie ma informacji na temat udziału doktoranta w konferencjach naukowych lub innych spotkaniach naukowych. Doktorant nie wykazał również żadnego artykułu, którego byłby głównym (pierwszym) autorem dotyczącego badań nad proponowaną metodyką.

Podsumowując należy stwierdzić, że dorobek publikacyjny Autora jest bardzo skromny jak na potrzeby pracy doktorskiej.

6. Czy autor wykazał umiejętność poprawnego i przekonującego przedstawienia uzyskanych przez siebie wyników /zwięzłość, jasność, poprawność redakcyjna rozprawy/?

Rozprawa napisana jest w poprawnym języku angielskim, sporadycznie zdarzają się błędy językowe.

Natomiast struktura pracy nie do końca jest prawidłowa. Można odnieść wrażenie, że Autor ma problem z przekazaniem informacji w jasny i spójny sposób. Tytuły głównych rozdziałów są bardzo krótkie i nie dostarczają treści wymaganej do zrozumienia ich zawartości, co znacznie utrudnia czytanie i zrozumienie pracy, np. rozdział 4. „Dissertation”, 5. „Functionalities”, 6. „Absent features”, 7. Implementation, itd.

Tytuł rozdziału 8 „Real use case” jest również mylący. Sugeruje, że w rozdziale znajdzie się praktyczny przykład opracowanej metodyki, wyniki z przeprowadzonych badań oraz wnioski. Natomiast w rozdziale zawarto jedynie informację, że opracowany prototyp kompilatora języka FBDL oraz zaproponowana przez Autora metodyka została wykorzystana przez firmę Fluence SP. z o.o. Ze względu na autorski charakter projektu nie można ujawniać żadnych szczegółów wewnętrznych. W załączniku C znajduje się oświadczenie firmy Fluence potwierdzające użycie kompilatora FBDL. Taki rozdział nie wnosi żadnej wartości merytorycznej.

Rodzi się zatem pytanie, gdzie w pracy można znaleźć wyniki przeprowadzonych prac badawczych, wyciągnięte wnioski, dyskusję, czy też dowody słuszności postawionej tezy?

W pracy nie ma dobrego przykładu pokazującego użycie opracowanej przez Autora metodyki i narzędzi do implementacji rejestrów oraz magistrali. Nie ma analizy oraz dyskusji uzyskanych wyników, dowodów potwierdzających słuszności postawionej tezy oraz porównania z istniejącymi narzędziami. Jeżeli projekt wykonany dla firmy Fluence SP. z o.o. jest objęty umową o zachowaniu poufności może warto wykorzystać prace wykonane w ramach eksperymentu CBM realizowanego wspólnie z ośrodkiem GSI Helmholtzzentrum für Schwerionenforschung, o którym wspomina Autor we wstępie pracy i na podstawie tego projektu przeprowadzić część badawczo-naukową oraz zademonstrować metodykę projektowania systemów bazujących na funkcjonalnym opisie rejestrów.

W pracy nie znalazłem odpowiedzi na powyższe pytania. Poprosiłem Doktoranta o wygłoszenie prezentacji na seminarium Katedry Mikroelektroniki i Technik Informatycznych Politechniki Łódzkiej w dniu 10 października, zaprezentowanie i omówienie jednego pełnego przykładu pokazującego użycie opracowanej metody. Uzyskałem dodatkowe informacje pozwalające mi lepiej zrozumieć wyniki pracy oraz braki występujące w monografii. Czytając ponownie pracę można dostrzec, że częściowe wyniki oraz wnioski są umieszczone bardzo chaotycznie w wszystkich rozdziałach od 1 do 8.

Praca wymaga poprawek mających na celu poprawienie jej struktury, tytułów rozdziałów, ich zawartości oraz dowodów postawionej tezy. W pracy należy zamieścić 1 lub 2 przykłady pokazujące wykorzystanie opracowanej metodyki oraz języka FBDL do implementacji rejestrów dołączonych do systemu przy pomocy magistrali. Należy również dokładniej opisać część dotyczącą samej magistrali, jej opisu, struktury oraz generacji programów dla układu programowalnego i procesora, zamieścić definicję parametrów magistrali, które można poznać bazując na opisie funkcjonalnym. Należy również zamieścić wnioski z przeprowadzonych badań oraz dowody słuszności postawionej tezy.

Obecnie wykonana przez Autora praca sprowadziła się do opracowania języka oraz napisania prototypu kompilatora, co jest zadaniem czysto inżynierskim. Natomiast, od Doktoranta wymaga się rozwiązania problemu w sposób naukowy i takie aspekty również w pracy powinny się pojawić. W pracy powinny zostać zawarte wyniki przeprowadzonych badań, eksperymentów oraz wnioski z nich wyciągnięte.

W pracy brakuje schematów blokowych opisywanych systemów, bloków cyfrowych pokazujących ich hierarchię, zależności logiczne oraz połączenia (magistrale), co utrudnia analizę jak również zrozumienie zależności pomiędzy układami. W wielu Autor zamieszcza wyłącznie wybrane listingi programów przedstawione w różnych językach programowania. Natomiast zamieszczenie wysokopoziomowych schematów blokowych znacznie ułatwiło by analizę przedstawionych układów cyfrowych, np. rozdział 1, gdzie Autor opisuje przykładowy problem systemu złożonego z bloków: Supervisor i Workers, natomiast zamieszczono wyłącznie programy dla modułu zarządzającego. Schemat blokowy powinien pokazać wymagane moduły oraz połączenia między nimi, rejestry oraz problemy, które Autor opisuje w rozdziale.

W załączniku A (Supervisor registerification results) zamieszczono plik opisujący wygenerowany plik rejestrów. Tutaj również zamieszczenie mapy rejestrów wraz z przypisanymi adresami względnymi oraz atrybutami, adresem bazowym znacznie ułatwiłoby analizę.

W pracy występują niepoprawnie sformatowane odwołania do literatury z licznymi brakami oraz niekompletną listą autorów uniemożliwiającymi sprawne odnalezienie publikacji.

7. Jakie są słabe strony rozprawy i jej główne wady?

W przedstawionej do recenzji pracy występują dwa główne problemy:

1. Brak dobrych przykładów użycia opracowanego języka FBDL do projektowania interfejsu bazującego na magistrali oraz rejestrach, jak również naukowych dowodów potwierdzających słuszność postawionej tezy.
2. Brak szczegółów dotyczących programów wygenerowanych przez kompilator języka FBDL (w języku opisu sprzętu) oraz języku wysokiego poziomu.

Prosiłbym zatem Autora o opracowanie przynajmniej jednego przykładu demonstrującego pełen proces implementacji rejestrów z użyciem języka FBDL: od napisania pliku fbd, poprzez proces kompilacji (front-end, back-end) do uzyskania plików wyjściowych. Przykłady powinny obejmować podstawowe mechanizmy operujące na rejestrach różnego typu: R, RW, RMW, wektor kilku rejestrów, moduł logiczny Proc, Config, Mask, Param. Bardziej skomplikowane moduły (pamięci i IRQ) nie są wymagane. Przykłady powinny być kompletne i obejmować rysunki przedstawiające magistralę systemową (dokładne informacje obejmujące sygnały oraz transfery), wymagany interfejs w postaci rejestrów (rysunek rejestrów wraz z adresami oraz dodatkowymi informacjami), omówienie rezultatów kompilacji, min.: plików wyjściowych w języku opisu sprzętu (np. VHDL lub Verilog) oraz programu dla procesora (np. C, C++, Python). Proszę również zamieścić kody pośrednie wygenerowane przez kompilator podczas tworzenia rejestrów. Idealnym demonstratorem byłoby pokazanie przykładu bazującego na praktycznym projekcie, np. eksperymencie CBM prowadzonym w ośrodku GSI, w którym Autor brał udział. Przedstawiony przykład powinien bezpośrednio odnosić się do postawionej tezy oraz dostarczać informacji dowodzących słuszności postawionej tezy.

W pracy brakuje właściwej części badawczej, badań eksperymentalne oraz analizy wyników dowodzącej słuszności postawionych tez. W pracy Autor głównie posługuje się subiektywnym kryterium oceny wyników przeprowadzonych badań, bazuje na przemyśleniach i rozważaniach. Czytając pracę można odczuć inżynierskie podejście do rozwiązania postawionego problemu. W przypadku pracy doktorskiej problem powinien zostać rozwiązany nie tylko w sposób inżynierski, ale również w sposób naukowy poprzez prowadzone badania oraz wyciągnięte wnioski.

Kierownik projektu RapidFAB stwierdził, że zastosowanie funkcjonalnie-zorientowanego podejścia pozwoliło skrócić czas implementacji oraz zwiększyć łatwość konserwacji systemu w porównaniu do poprzednio stosowanego podejścia bazującego na rejestrach. Natomiast brakuje podstawowych informacji na temat wykonanego projektu, stopnia skomplikowania, użytych układów programowalnych oraz metodyki i kryteriów których użyto do pomiaru wydajności oraz analizy utrzymania systemu. Trudno jest zatem ocenić naukowe aspekty wdrożenia.

W pracy brakuje usystematyzowanego kryterium oceny wyników badań dowodzących przydatności zaproponowanego języka, odniesienia do aktualnego stanu wiedzy oraz jawnych dowodów zaproponowanej tezy. W pracy brakuje również wyników eksperymentalnych badań z użyciem języka FBDL, np.: o ile szybciej, lepiej udało się zrealizować implementację rejestrów w omawianych przedstawionych przykładach. Najlepiej jak Doktorant przedstawiłby metodykę jaką posłużył się w pracy do rozwiązania problemu naukowego.

W pracy nie zostało napisane jakie obiekty oraz mechanizmy opisane w rozdziale 5 zostały zaimplementowane i przebadane. Na potrzeby pracy doktorskiej nie ma potrzeby implementacji oraz zbadania wszystkich mechanizmów. Prosiłbym Doktoranta o wskazanie które mechanizmy zostały zaimplementowane oraz zbadane. Prosiłbym również o wyjaśnienie jakie języki są obecnie wspierane przez kompilator, zarówno do generacji programu Providera, jak również Requestera.

W pierwszym rozdziale Autor określił zadania dotyczące projektowania magistral obejmujące zarządzania magistralą adresową, danymi, strukturą magistrali (Bus fabric management). Prosiłbym o wyjaśnienie w jaki sposób zaproponowany język FBDL pozwala na opis oraz modyfikację magistrali systemowej i jej parametrów (np. zmiana szerokości magistrali adresowej, typu magistrali, adresów bazowych), jakie są ograniczenia. Proszę posłużyć się przykładami przygotowanymi w języku FBDL oraz wygenerowanym opisem układu cyfrowego w języku opisu sprzętu.

W rozdziale 5.2 Autor opisuje funkcjonalność projektowania magistral systemowych. Rozdział głównie skupia się na dyskusji dotyczącej konwencji nazewnictwa natomiast brakuje informacji dotyczących opisu samej magistrali. Występuje tu wyłącznie magistrala typu „Main”. Co prawda, trochę więcej informacji zawarty jest w rozdziale 7.2. „Bus” specyfikacji języka FBDL. Natomiast nadal są to szczątkowe informacje, które nie dostarczają dowodów na pierwszą część tezy. Warto byłoby opisać w rozdziale 5.2, które typy magistral są wspierane przez język oraz jak można poznać oraz zmienić parametry magistrali posługując się opisem funkcjonalnym danych. W rzeczywistych systemach mikroprocesorowych format danych przechowywanych w rejestrach nie ma związku z rodzajem magistral wykorzystywanych do transmisji danych (magistrala adresowa, danych), np. zapis 32-bitowego rejestru może odbywać się z użyciem 8-bitowej magistrali danych i 24-bitowej magistrali adresowej. Prosiłbym o wyjaśnienie w jaki sposób można poznać strukturę magistrali systemowej bazując na opisie funkcjonalnym danych. Opis dotyczący

projektowania magistral dotyczy bezpośrednio pierwszej części tezy. Przedstawiony opis powinien dowodzić jej słuszności w przekonujący sposób.

Autor nie wspomina również o metodach stosowanych w systemach mikroprocesorowych pozwalających na implementację rejestrów w postaci struktury, unii (język C), czy też bardziej złożonych obiektów (język C++). Czy takie elementy były przez Autora rozważane podczas prac nad językiem FBDL?

W pracy dostrzeżono pewną liczbę błędów, głównie edytorskich, które nie mają znaczącego wpływu na wartość merytoryczną pracy doktorskiej. W pracy występują liczne niepoprawnie opisane i sformatowane odwołania do literatury z licznymi brakami uniemożliwiającymi odnalezienie publikacji, np. [79] - braki kapitalizacji, [9] – brak większości wymaganych pól., itd. W niektórych przypadkach niemożliwe jest nawet odnalezienie publikacji, np.: pozycja [53] odnosi się wyłącznie do nazwy ośrodka „Deutsches elektronen-synchrotron” (również napisanego błędnie). Taka referencja nie wnosi żadnej merytorycznej treści. Niewłaściwie zostały opisane referencje, np. [17] – tutaj autorem jest konsorcjum PCI-SIG (Peripheral Component Interconnect Special Interest Group), Autor używa skrótu P. SIG co sugeruje imię i nazwisko autora. W wielu odnośnikach brakuje kapitalizacji w tytule „Pci specification”, „Isa bus specification...”, „Vesa local bus...”, itd.

W pracy brakuje definicji niektórych akronimów i skrótów, np. SCSI, PCI na stronie 24, itd.

Na listingu 5 pole WORKER_COUNT zawierać wartość większą niż 32 (obecnie jest 24).

Doktorant powinien również pamiętać, że praca realizowana jest w dyscyplinie Automatyka, Elektronika, Elektrotechnika i Technologie Kosmiczne (AEEiTK) i powinna dotyczyć elektroniki. Natomiast, opracowanie języka programowania oraz kompilatora wpisuje się w dyscyplinę Informatyka Techniczna i Telekomunikacja. Z tego względu wyniki prac powinny bezpośrednio dotyczyć dyscypliny AEEiTK. Prosiłbym zatem Doktoranta o dokładniejsze omówienie aspektów związanych z układem programowalnym obejmujących magistralę oraz logikę implementującą rejestry.

8. Jaka jest przydatność rozprawy dla nauk technicznych?

Na tym etapie, trudno jest jednoznacznie ocenić przydatność rozprawy dla nauk technicznych ze względu na braki dotyczące wyników badań naukowych oraz przykładów ilustrujących kompletny proces implementacji magistral oraz rejestrów w systemach cyfrowych.

Niemniej jednak należy podkreślić praktyczne znaczenie pracy dla nauk technicznych. Zaproponowane przez Autora funkcjonalne podejście dotyczące implementacji rejestrów (2-ga część tezy) w złożonych systemach cyfrowych może znacznie skrócić oraz ułatwić nie tylko proces projektowania takich układów, ale również utrzymania oprogramowania. Sama metodyka może być pomocna na dalszych etapach implementacji, czy też rozbudowy takich układów, kiedy w już istniejącym oprogramowaniu należy zaimplementować nowe rejestry lub zmodyfikować już istniejące. Na tym etapie zwykle następuje „rosynchronizowanie” projektu opisującego logikę układu programowalnego oraz bibliotek napisanych w językach wysokiego poziomu dostarczających funkcje do komunikacji z logiką układu, co generuje błędy w działaniu układu, powoduje opóźnienia w rozwoju takich systemów oraz wymaga ponownego procesu dokładnego testowania całego systemu.

Opracowany prototyp kompilatora języka FBDL oraz zaproponowana przez Autora metodyka została wykorzystana przez firmę Fluence SP. z o.o. podczas opracowania modułu generatora opóźnień dla femtosekundowego lasera realizowanego w ramach

projektu „Opracowanie układu optycznego do szybkiej laserowej obróbki materiałów przezroczystych” (akronim: RapidFAB), co jest bardzo dobrym przykładem praktycznego zastosowania rezultatów badań. Szkoda tylko, że brakuje szczegółów dotyczących tego projektu.

Rezultaty opracowanej metodyki mogą zostać wykorzystane nie tylko w przemyśle, a przede wszystkim w eksperymentach wielkiej skali takich, jak choćby, wspomniany przez Autora eksperyment projektu badawczego CBM realizowany w GSI – instytucie Badań Ciężkich Jonów w Darmstadt, Niemcy.

Rezultaty badań mogą zostać również wykorzystane podczas projektowania dedykowanych układów scalonych ASIC (Application Specific Integrated Circuit), w których istnieje potrzeba implementacji interfejsu sprzęt-oprogramowanie.

9. Do której z następujących kategorii Recenzent zalicza rozprawę:

- a) nie spełnia wymagań stawianych rozprawom doktorskim przez obowiązujące przepisy
- b) **wymaga wprowadzenia poprawek i ponownego recenzowania**
- c) spełnia wymagania
- d) spełnia wymagania w wyraźnym nadmiarze
- e) wybitnie dobra, zasługująca na wyróżnienie

Autor zaproponował nową metodykę projektowania systemów cyfrowych wykorzystując opis funkcjonalny danych oraz obiektów stanowiących interfejs pomiędzy układem programowalnym, a oprogramowaniem wysokiego poziomu procesora. Cel oraz problem zostały prawidłowo określone, natomiast w pracy brakuje wyników badań oraz twardych dowodów dowodzących słuszność postawionej tezy. Prace realizowane są w dyscyplinie naukowej Automatyka, Elektronika, Elektrotechnika i Technologie Kosmiczne i badania naukowe powinny być realizowane w tym obszarze.

W podsumowaniu stwierdzam, że praca doktorska mgr inż. Michała Kruszewskiego pt. „Język Opisu Funkcjonalnych Magistral”, napisana w dyscyplinie elektronika Automatyka, Elektronika, Elektrotechnika i Technologie Kosmiczne wymaga wprowadzenia poprawek zgodnie z zawartymi wnioskami i sugestiami w punktach 3 – 8 recenzji.

Podpis

