

WPŁYNEŁO  
2024 -04- 25  
dn.....

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

Łódź, dn. 08.04.2024 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 oraz pisma z dnia 04.03.2024 r. opracowanego na podstawie uchwały nr 699/II/2024 r. dotyczącej zgody na uzupełnienie i poprawę rozprawy doktorskiej w postępowaniu w sprawie nadania stopnia doktora mgr inż. Michała Kruszewskiego.

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 system komputerowy. Systemy wykorzystujące procesory oraz układy programowalne stają się coraz bardziej popularne ze względu na ich duże możliwości obliczeniowe oraz coraz niższe koszty ich wytworzenia.

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 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 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. W pracy zawarto przykład pokazujący użycie języka FBDL do automatycznej implementacji rejestrów na podstawie opisu funkcjonalnego, porównanie z generatorem AGWB oraz analizę zalet takiego podejścia.

Zagadnienie naukowe omawiane w pracy zostało sformułowane w postaci jednej tezy zamieszczonej w rozdziale 4.1 na stronie 58 (tłumaczenie polskie oraz wersja oryginalna w jęz. angielskim):

*„Możliwe jest wygenerowanie opisu sprzętowego struktury rejestrów oraz programowych metod dostępu do danych 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 generate a hardware description of the bus register structure and software data access methods based on the description of the functionality of the data that shall be stored in the registers. Moreover, such an approach offers some advantages in certain practical 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 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 oraz porównanie z klasycznym podejściem zorientowanym na rejestry.

**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 10 rozdziałów poprzedzonych wykazem skrótów, krótkim wprowadzeniem, wykazem literatury oraz ośmioma załącznikami. W załączniku 8 Autor zamieścił opis opracowanego języka FBDL (45 stron).

Układ merytoryczny pracy obejmuje część teoretyczną oraz praktyczną składającą się z opisu specyfikacji funkcjonalnej, implementacji kompilatora oraz przykładu obrazującego implementację rejestrów z użyciem języka FBDL. 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 metod oraz 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. W podsumowaniu 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 oraz przedstawia pojedynczą tezę badawczą. Celem badań naukowych jest opracowanie oraz zbadanie języka umożliwiającego opis 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: podukładów cyfrowych (Blackbox), 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 Autor zamieścił dwa przykłady projektów ilustrujących cały proces implementacji rejestrów z użyciem klasycznego podejścia rejestrowego (narzędzie AGWB) oraz bazującego na funkcjonalnym opisie rejestrów przy pomocy języka programowania FBDL. W obu przypadkach Autor wykorzystał bibliotekę języka VHDL dedykowaną dla magistrali Wishbone, środowisko symulacyjne GHDL oraz skrypty napisane w języku Python. W przykładzie zaimplementowano 3 rejestry konfiguracyjne, 3 rejestry statusowe o różnej liczbie bitów, macierze 10 rejestrów sterujących i statusowych, rejestr 33 bitowego

licznika, rejestr maski, statyczny rejestr wersji oprogramowania oraz dwa bloki sumatorów. Autor zamieścił listingi programów pokazujące implementację z użyciem generatora AGWB oraz języka FBDL. Programy użyte podczas symulacji dostępne są w repozytorium. Oba podejścia zostały zweryfikowane przy pomocy symulacji z użyciem programów napisanych w językach VHDL oraz Python. W załącznikach A-F zamieszczono wyniki pośredniego procesu generacji rejestrów, mapę wygenerowanych rejestrów, programy i funkcje w języku Python dedykowane do obsługi rejestrów oraz programy w języku VHDL dla 2 modułów (Main oraz Subblock). W rozdziale Autor przedstawił zalety podejścia bazującego na opisie funkcjonalnym oraz określił 4 kryteria pozwalające na porównanie obu rozwiązań. W dalszej części dokonał porównania wygenerowanych rejestrów dla różnych przypadków rejestrów, macierzy rejestrów, przypadków z dostępem atomowym do rejestrów dłuższych niż magistrała danych, bloków cyfrowych z dostępem strumieniowym oraz rejestrów masek bitowych. W ostatniej części zamieszczono wyniki syntezy obu rozwiązań dla układu FPGA Xilinx z rodziny Spartan 7 (XC7S25) dostępnego na płycie prototypowej Cmod S7. Podejście bazujące na opisie funkcjonalnym pozwoliło na zmniejszenie zasobów układu programowalnego o około 25% w przypadku obu modułów. Optymalizacja występuje głównie w logice sterującej dostępem do rejestrów ze względu na bardziej efektywne rozłożenie rejestrów oraz optymalizację dekodera adresowego.

W rozdziale 9 „Praktyczne użycie” Autor zamieścił 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 10 Autor podsumowuje zalety zaproponowanego podejścia funkcjonalnego oraz języka FBDL w odniesieniu do klasycznego podejścia bazującego na rejestrach.

Autor odwołuje się w pracy do 108 pozycji literaturowych. W pracy występuje niewielka liczba odwołań do źródeł naukowych wnoszących wartość naukową do pracy (5 pozycji książkowych, 38 artykuły naukowe). Około połowy stanowią artykuły pochodzące z czasopism z listy filadelfijskiej. Natomiast ponad 51% stanowią odwołania do źródeł elektronicznych udostępnionych przez firmy oraz ośrodki naukowo-badawcze obejmujących: repozytoria Git (19 pozycji), 2 raporty, 1 prezentacja, 10 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 (51%) 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.

Wśród cytowanych prac znajdują się 3 pozycje współautorstwa Doktoranta: Sensors/MDPI (IF=3,9), Electronics/MDPI (IF=2,9) oraz Photonics Applications in Astronomy, Communications, Industry and High-Energy Physics Experiments/International Society for Optics and Photonics. Autor zamieścił również odwołanie do 4 repozytoriów na których umieścił fragmenty kompilatora. W artykule Electronics/MDPI doktorant jest jedynym autorem. W artykule opisano rezultaty badań przeprowadzonych w doktoracie.

#### **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 rejestrów poprzez zdefiniowanie typu przechowanych w nich danych. Praca obejmowała również opracowanie prototypu kompilatora pozwalającego na zweryfikowanie poprawności proponowanego rozwiązania, zbadanie oraz porównanie zaproponowanej metodyki bazującej na opisie funkcjonalnym rejestrów z rozwiązaniami klasycznymi.

W pracy Autor nie opisał jawnie 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. Rezultaty badań zostały opisane w rozdziale 8 „Example design”, w którym Autor zaprezentował dwa przykłady implementacji złożonych rejestrów oraz programów operujących na rejestrach z wykorzystaniem opracowanego języka FBDL oraz narzędzia AGWB. W celu porównania obu rozwiązań Autor posłużył się środowiskiem symulującym system mikroprocesorowy (program testowy oraz biblioteki zapewniające dostęp do rejestrów opracowane w języku Python) jak również logikę cyfrową symulującą dwa układy wykorzystujące rejestry (Main.vhd oraz Subblock.vhd) dołączone przy pomocy magistrali Wishbone. Symulacje układu cyfrowego przeprowadził z użyciem symulatora GHDL 3.0.0, natomiast układy cyfrowe zostały opisane w języku VHDL. W rozdziale 8 Autor zawarł cenne informacje, których brakowało w pierwszej wersji pracy, m.in.: spójny przykład ilustrujący cały proces implementacji rejestrów, informacje dotyczące logiki zawartej w układzie programowalnym, strukturę rejestrów, przypisanie adresów do rejestrów, połączenie bloku rejestrów z magistralą oraz implementacji bardziej złożonych obiektów oraz równoległy przykład obrazujący proces implementacji rejestrów z użyciem klasycznych rozwiązań. Autor opracował 4 kryteria pozwalające na porównanie obu rozwiązań obejmujące: utrzymanie kodu (maintability), czytelność programu (readability), prawdopodobieństwo popełnienia błędu (safety) oraz czas (time) potrzebny na implementację programów realizujących oraz zarządzających rejestrami, co pozwoliło na porównanie obu metod oraz pokazanie zalet zaproponowanego rozwiązania. Użycie języka FBDL pozwoliło na wyeliminowanie klasycznych problemów, z którymi spotyka się



programista podczas implementacji lub zmiany rejestrów w złożonych systemach cyfrowych, np. zmiany długości rejestrów, zmiany architektury rejestrów, przesunięcia lub zmiany adresów, operacje dostępu atomowego, modyfikacje bloków rejestrów, czy też bardziej złożonych układów cyfrowych. W większości przypadków zmiany realizowane są przez kompilator języka FBDL, co upraszcza proces modyfikacji programu, zmniejsza prawdopodobieństwo wystąpienia błędów podczas programowania oraz zwalnia programistę z mozolnego obliczania adresów rejestrów oraz masek bitowych.

W rozdziale 8.2 Autor opisał badania przeprowadzone z użyciem płyty ewaluacyjnej z układem FPGA AMD Spartan 7, w którym zaimplementowano rejestry, magistrale Wishbone. System komputerowy został dołączony przy pomocy interfejsu SPI oraz mostka SPI-Wishbone. Takie podejście pozwoliło na implementację oraz porównanie obu rozwiązań w rzeczywistym układzie programowalnym oraz analizę wykorzystanych zasobów sprzętowych układu FPGA.

Zaproponowany przez Autora język FBDL pozwala na funkcjonalne opisanie rejestrów. Autor zamieścił przykłady oraz wyniki symulacji i implantacji w układzie FPGA dowodzące użyteczności zaproponowanej metodyki implementacji rejestrów.

**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 H.

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 kompilatora składający się z dwóch elementów: front-end oraz back-end. Opis wspieranych funkcjonalności zawarty jest w rozdziale 7. Opracowane narzędzia zostały zbadane oraz przetestowane z użyciem środowiska symulacyjnego jak również układu programowalnego.

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). Na repozytorium można również znaleźć kody programów wykorzystywanych podczas badań nad doktoratem.

Doktorant jest współautorem 3 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). Natomiast trzecia autorska publikacja została opublikowana w czasopiśmie MDPI, Electronics w 2024 roku.

W pracy doktorskiej nie ma informacji na temat udziału doktoranta w konferencjach naukowych lub innych spotkaniach naukowych.

Podsumowując należy stwierdzić, że dorobek publikacyjny Autora jest wystarczający 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”, rozdział 8 “Example design”, itd. Tytuł rozdziału 9 „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 G znajduje się oświadczenie firmy Fluence potwierdzające użycie kompilatora FBDL podczas opracowania modułu generatora opóźnień. Taki rozdział nie wnosi żadnej wartości merytorycznej.

Wyniki przeprowadzonych badań w ramach doktoratu zostały opisane w rozdziale 8 „Example design”. Tytuł rozdziału jest również niewłaściwy. W rozdziale nie tylko zawarto opis dwóch przykładów, ale również cenne informacje obejmujące: opis wyników badań, 4 kryteria pozwalające na porównanie rejestrów zaimplementowanych z użyciem języka FBDL oraz generatora AGWB, jak również, wnioski oraz dyskusję dowodzącą słuszności postawionej tezy. W rozdziałach 8.1.1 – 8.1.5 Autor posługuje się skrótami w nawiasach (MT, MRT, MST, MRS, R), które chyba powinny się mapować na zdefiniowane kryteria. Natomiast skróty nie są w pracy wyjaśnione i nie zgadzają się ze skrótami kryteriów podanych w rozdziale 8.1 (M, R, S, T), co znacznie utrudnia czytanie pracy.

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

Pomimo wyżej wymienionych problemów praca opisuje użyteczny język programowania oraz wyniki badań, porównania i trafne wnioski pozwalające na bardziej wydajną implementację oraz zarządzanie rejestrami w układach cyfrowych. Rezultaty badań oraz wnioski zostały przedstawione w sposób przejrzysty i przekonujący, aczkolwiek są trudne do odnalezienia w pracy.

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

W pierwszej wersji pracy przedstawionej do recenzji występowały 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.

Oba problemy zostały rozwiązane w obecnej wersji pracy. Doktorant przedstawił również odpowiedzi na pytania oraz inne problemy poruszone w poprzedniej recenzji.

W przedstawionej do recenzji pracy dostrzeżono pewną liczbę błędów, głównie edytorskich, które nie mają znaczącego wpływu na wartość merytoryczną pracy doktorskiej. W pracy nadal występują liczne niepoprawnie opisane i sformatowane odwołania do literatury z licznymi brakami uniemożliwiającymi odnalezienie publikacji, np. [34], [52] - brak informacji dotyczących czasopisma, wydawcy lub konferencji. Podobnie braki występują w referencji [85]. W niektórych przypadkach niemożliwe jest nawet odnalezienie publikacji, np.: pozycja [58] odnosi się wyłącznie do nazwy ośrodka „Deutsches elektronensynchrotron”. Taka referencja nie wnosi żadnej merytorycznej treści. Podobne uwagi zostały zgłoszone do pierwszej wersji pracy. Pomimo tego Autor nadal niepoprawnie opisuje odwołania do literatury.

W rozdziałach 8.1.1 – 8.1.5 Autor posługuje się skrótami w nawiasach (MT, MRT, MST, MRS, R), które chyba powinny się mapować na zdefiniowane wcześniej kryteria. Natomiast skróty nie są w pracy wyjaśnione i nie zgadzają się ze skrótami kryteriów podanych w rozdziale 8.1 (M, R, S, T), co znacznie utrudnia czytanie pracy. Prosiłbym Autora o wyjaśnienie skrótów MT, MRT, MST, MRS, R oraz przedstawienie poprawnego mapowania na zdefiniowane kryteria oceny.

Brak numerów rozdziałów w zakładce pliku pdf znacznie utrudnia nawigację oraz czytanie pracy. W obecnych czasach standardem jest posługiwanie się dokumentami w wersji elektronicznej, więc również ważne jest dopracowanie tego dokumentu.

W pracy autor wspomina o praktycznym wykorzystaniu opracowanego języka FBDL podczas realizacji projektu realizowanego w ramach eksperymentu CBM realizowanego wspólnie z ośrodkiem GSI. Prosiłbym doktoranta o zwięźle omówienie (2-3 slajdy) wkładu w ten projekt dotyczącego wykorzystania języka FBDL.

## **8. Jaka jest przydatność rozprawy dla nauk technicznych?**

Należy podkreślić praktyczne znaczenie pracy dla nauk technicznych. Zaproponowane przez Autora funkcjonalne podejście dotyczące implementacji rejestrów w złożonych systemach cyfrowych może znacznie skrócić oraz ułatwić nie tylko proces projektowania takich układów, ale również utrzymania oprogramowania, czy też debugowania programów. 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 „rozsynchronizowanie” (zmiany adresów rejestrów oraz masek bitowych) 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. Przedstawione przez Autora przykłady w rozdziale 8 doskonale pokazują zalety zaproponowanego języka i zachęcają do jego użycia. Pewnym

mankamentem jest brak kompilatora języka FBDL, ale może w przyszłości pojawi się projekt kompilatora rozwijanego i utrzymywanego przez większą liczbę programistów.

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 nadmiarem
- e) wybitnie dobra, zasługująca na wyróżnienie

Rozprawa mgr inż. Michała Kruszewskiego ma charakter praktyczno-doświadczalny. 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. Autor przeprowadził badania nad kompilatorem języka FBDL oraz opracował kryteria i metody pozwalające na porównanie zaproponowanej metodyki z istniejącymi rozwiązaniami (generator AGWB) oraz osiągnął założony cel.

W podsumowaniu stwierdzam, że rozprawa doktorska mgr inż. Michała Kruszewskiego pt. „Język Opisu Funkcjonalnych Magistral”, napisana w dyscyplinie automatyka, elektronika, elektrotechnika i technologie kosmiczne, spełnia wymagania określone w Ustawie z dnia 20 lipca 2018 r. Prawo o szkolnictwie wyższym i nauce (Dz. U. z 2023 r., poz. 742 z późniejszymi zmianami). Wnioskuje, zatem o dopuszczenie rozprawy do publicznej obrony.

Podpis

