

Prof. dr hab. inż. Lucyna Leniowska
Zakład Mechatroniki i Automatyki
Instytut Informatyki
Uniwersytet Rzeszowski
ul. Prof. S. Pigonia 1, 35-310 Rzeszów

Rzeszów, dn. 29.05.2026 r.

Recenzja pracy doktorskiej

pt.: Wielordzeniowa maszyna wirtualna dla sterowników programowalnych

Autor rozprawy: mgr inż. Marcin Hubacz

Promotor rozprawy: Prof. dr hab. inż. Leszek Trybus

Promotor pomocniczy: dr inż. Bartosz Trybus

Dyscyplina: Informatyka Techniczna i Telekomunikacja

1. Podstawa prawna

Recenzja pracy doktorskiej została przygotowana na podstawie decyzji Rady Dyscypliny *Informatyka Techniczna i Telekomunikacja*, w oparciu o otrzymany egzemplarz rozprawy doktorskiej, przesłanej wraz z pismem przewodnim Przewodniczącego Rady Dyscypliny, Pana dra hab. inż. Mariusza Oszusta, prof. PRz. Recenzja została opracowana zgodnie z Ustawą z dnia 20 lipca 2018 r. Prawo o szkolnictwie wyższym i nauce (Dz. U. z 2018 poz. 1668 z późniejszymi zmianami).

2. Ogólna charakterystyka pracy

Przedłożona do oceny rozprawa doktorska podejmuje temat rozszerzenia możliwości maszyny wirtualnej, przeznaczonej dla środowiska jednorodzeniowego na środowisko wielordzeniowe, na potrzeby badania i konstrukcji układów automatyki przemysłowej. Problem ten jest istotny zarówno z poznawczego jak i z aplikacyjnego punktu widzenia, zaś podstawę pracy stanowi środowisko CPDev, rozwijane na Politechnice Rzeszowskiej.

Dysertacja napisana jest w języku polskim, zawiera łącznie 126 stron i składa się z 7 rozdziałów, poświęconych kolejno:

Rozdział pierwszy zawiera wprowadzenie do tematu, cel i zakres badań. Cel jest dobrze umotywowany. Autor objaśnia problem, który powstał w związku z pojawieniem się procesorów wielordzeniowych, stosowanych także w nowoczesnych sterownikach.

W przypadku większej liczby rdzeni można jednocześnie wykonywać więcej zadań, co spowodowało konieczność unowocześnienia stworzonego na Politechnice Rzeszowskiej środowiska CPDev i rozszerzenia go na architekturę wielordzeniową.

W rozdziale drugim Autor omawia szczegóły środowiska CPDev, szczególnie jego część odpowiedzialną za kompilację programów. Opisuje m. in. proces tłumaczenia programów z języków zgodnych z normą IEC 61131-3 do pośredniego języka VMASM, a następnie do kodu wykonywalnego.

Rozdział trzeci Autor poświęcił maszynie wirtualnej, stanowiącej część wykonawczą środowiska CPDev. Opisał jej architekturę, obejmującą moduł wykonujący instrukcje VMASM, pamięci kodu i danych, stosy, rejestry oraz interfejs sprzętowy. Proces wykonywania instrukcji został przedstawiony zarówno w formie algorytmów, jak i modeli denotacyjnych, ułatwiających implementację w językach C/C++. Omówione zostały również narzędzia do uruchamiania i debugowania programów, takie jak symulator *CPSim* i nowe środowisko IDE.

W rozdziale czwartym Doktorant analizuje wpływ różnych sposobów dostępu do pamięci na wydajność maszyn wirtualnych 16- i 32-bitowych: VM16, VM32 i VM32A. Porównuje metody dostępu bajtowego, wykorzystania funkcji *memcpy* oraz bezpośredniego dostępu za pomocą wskaźników. Wyniki pokazały, że odpowiedni sposób dostępu do pamięci może znacząco zwiększyć szybkość działania maszyn wirtualnych, jednak często odbywa się to kosztem większego rozmiaru kodu programu.

W rozdziale piątym Autor przedstawia koncepcję rozszerzenia środowiska przeznaczonego dotąd dla pojedynczego zadania na system dwurdzeniowy, realizujący równolegle dwa zadania. Zamiast wykorzystania systemu operacyjnego RTOS, Doktorant proponuje komunikację między rdzeniami za pomocą pamięci współdzielonej, zaś wymianę danych planuje poprzez zmienne globalne zgodne z normą IEC 61131-3.

W rozdziale 6 opisano rozszerzenie środowiska CPDev do pracy na procesorach dwurdzeniowych oraz jego praktyczną implementację na mikrokontrolerze STM32H755. Z punktu widzenia użytkownika zmiany w środowisku są niewielkie i dotyczą głównie dodania atrybutów READ/WRITE dla zmiennych wymienianych między rdzeniami. Większe modyfikacje objęły język VMASM oraz maszynę wirtualną, które rozszerzono o mechanizmy

bezpiecznej komunikacji przez pamięć współdzieloną. Poprawność działania sprawdzono w symulacjach oraz na platformie sprzętowej STM32.

W rozdziale 7 przedstawiono bardziej zaawansowany system wykorzystujący identyfikację RFID oraz robota mobilnego sterowanego przez procesory ESP32 i *Raspberry Pi*. System odpowiadał za wyszukiwanie i odczyt znaczników RFID rozmieszczonych na planszy. Rdzenie *Raspberry Pi* realizowały logikę sterowania i komunikację, natomiast ESP32 odpowiadał za sterowanie ruchem robota. Program sterujący został napisany w języku ST i został podzielony na współpracujące podprojekty, komunikujące się przez pamięć współdzieloną i Wi-Fi. Opisano także sposób działania systemu w sytuacjach błędnych, np. gdy robot zgubi wyznaczoną trasę.

Rozprawę uzupełnia **Podsumowanie** i spis literatury, który liczy 88 pozycji. Literatura przedmiotu cytowana przez Autora jest właściwie dobrana i w większości anglojęzyczna. Układ pracy jest poprawny. Treść rozprawy została logicznie podzielona na rozdziały, a ich zawartość i zakres nie budzi zastrzeżeń.

3. Oryginalne osiągnięcia pracy

Niezwykle szybki rozwój mikrokontrolerów ARM, wraz z różnorodnymi układami peryferyjnymi (w tym transmisji przewodowej i bezprzewodowej), połączony z masową ich produkcją, wygenerował istotne zmiany w wielu gałęziach gospodarki, między innymi w dziedzinie automatyki przemysłowej. Pojawienie się wielordzeniowych układów ARM o różnych konfiguracjach rdzeni, skomplikowało znacznie proces tworzenia dla nich oprogramowania. Dotyczy to zarówno oprogramowania narzędziowego, bibliotek wspomagających oraz oprogramowania aplikacyjnego. Aby uniknąć chaosu, zwłaszcza w sektorze automatyki przemysłowej, opracowano normy standaryzujące aplikacje systemów automatyki w kilku kluczowych aspektach obejmujących: unifikację języków programowania, standardy wymiany informacji, reguły projektowania systemów sterowania oraz ustalenie zbiorów wymagań dotyczących części sprzętowej sterownika.

Jednak pomimo pojawienia się norm standaryzujących projektowanie układów automatyki, problem podstawowy, czyli konieczność opracowania oprogramowania dla dużej liczby różnych rodzajów kontrolerów występujących na rynku, nie zniknął. Jedno z możliwych rozwiązań tego problemu może polegać na wykorzystaniu środowiska programistycznego, wyposażonego w wirtualną maszynę, zdolną realizować zadania dla sterownika PLC oraz

regulatorów typu PID. Takie podejście zastosowano w budowaniu środowiska programistyczno-wykonawczego CPDev, które powstało w Katedrze Informatyki i Automatyki Politechniki Rzeszowskiej. W pełni funkcjonalne środowisko CPDev obejmuje: architekturę systemu, postać modelu komunikacyjnego, sposób konfiguracji oraz typy wykorzystywanych danych. System ten obsługuje tekstowe, graficzne i mieszane języki programowania (ST, IL, LD, FBD, SFC). Ważną częścią składową systemu CPDev jest język pośredni VMASM, który przetwarza kod ST oraz IL do formy assemblera, przekształcanego w kolejnym kroku na kod binarny.

Istotny mankament platformy z wirtualną maszyną uwidocznił się wraz z pojawieniem się sterowników wielordzeniowych, w szczególności wyposażonymi w rdzenie ARM o różnych specyfikacjach (serie A, M3, M4, M7, L, N, P, R, W). Dodatkowo, układy o większych możliwościach posiadają również jednostki GPU o kilkunastu rdzeniach, co naturalną kolejną rzeczą spowodowało potrzebę wykorzystania tych zasobów.

Praca doktorska Pana mgr. inż. Marcina Hubacza dotyczy zagadnień związanych z projektowaniem i implementacją wirtualnych sterowników programowalnych na wielordzeniowe procesory/kontrolery, na potrzeby badania i konstrukcji układów automatyki przemysłowej. Autor wykorzystuje denotacyjny model maszyny wirtualnej do opracowania jej implementacji w języku C/C++, wyposażonej w bezkolizyjny dostęp do pamięci współdzielonej. Jest to tematyka ważna na każdym etapie projektowania, wykonywania i uruchamiania złożonego systemu automatyki przemysłowej, a jego rozwiązanie jest niewątpliwie ważnym i poszukiwanym celem. Autor sformułował cel pracy następująco:

Głównym celem niniejszej pracy jest opracowanie metody rozszerzenia maszyny wirtualnej przeznaczonej dla środowiska jednordzeniowego na środowisko wielordzeniowe, w którym rdzenie wykonują programy IEC 61131-3 mogące wymieniać między sobą dane”.

Warto zaznaczyć, że przejście z systemu obsługującego jeden rdzeń do układu wielordzeniowego jest procesem trudnym, wymagającym posiadania unikalnych zdolności programistycznych i dogłębnego zrozumienia architektury układu. Pomimo dostępnej specyfikacji układu, jej praktyczne wykorzystanie jest możliwe dla wąskiej grupy specjalistów. W zakresie nowości - w rozumieniu naukowo-technicznym, należy wskazać rozszerzenie jednordzeniowej maszyny wirtualnej na model wielordzeniowy, oparty o wiele instancji maszyn wirtualnych współpracujących i wymieniających dane przez pamięć współdzieloną. W ramach tego podejścia rozwiązane zostały problemy związane z:

- formalizacją zasad wymiany danych między rdzeniami poprzez rozszerzone deklaracje zmiennych globalnych (READ/WRITE);

- synchronizacją wymiany danych, przez wprowadzenie zasady wymiany na początku i na końcu cyklu sterowania zamiast klasycznego planisty RTOS;
- rozszerzeniem modelu maszyny wirtualnej o współdzielenie pamięci i wielordzeniowość.

Doktorant dodatkowo przeprowadził analizę wpływu metod dostępu do pamięci na wydajność. Zastosowano cztery rodzaje technik dla różnej postaci danych (BYTE_ACCESS, MEMCPY_ACCESS, DIRECT_ACCESS). W kolejnym kroku badań opracowane schematy wymiany danych przez pamięć zostały zaimplementowane i sprawdzone na wielordzeniowych architekturach ESP32 (dwa identyczne rdzenie) oraz na STM327xx (CORTEX M7 + CORTEX M4).

Aby zabezpieczyć się przed błędami, jakie mogą wystąpić w trakcie projektowania metod obsługujących funkcjonalności oprogramowania rzutowanego na wielordzeniową architekturę, Doktorant wykorzystuje denotacyjny model maszyny wirtualnej. Model ten powinien być jednocześnie precyzyjny tak, aby nie komplikować języka, który ma go obsługiwać (funkcji), a jednocześnie powinien pozwalać na rozszerzenia w przyszłości. Wydaje się, że pierwsza własność modelu została spełniona, ale czy druga – pozwalająca na dalsze rozszerzenia także?

Doktorant sprawnie porusza się w podjętej tematyce, wykazując się wiedzą, zarówno w obszarze architektury układów mikroprocesorowych i sterowników programowalnych, jak i w zakresie ich programowania. Zaletą pracy jest wykonanie eksperymentów, które potwierdzają użyteczność i funkcjonalność stworzonego rozwiązania.

4. Uwagi krytyczne i dyskusyjne

Po przeczytaniu pracy nasunęły mi się następujące uwagi krytyczne lub dyskusyjne:

1. Doktorant stosuje bezosobową formę opisu poszczególnych etapów badań i nie jest wystarczająco jasne, co jest własnym wkładem Autora, a co zostało zaczerpnięte z prac innych badaczy?
2. Obecnie na rynku mamy układy ARM o wielu rdzeniach i kilku architekturach, np. posiadających dwie magistrale pamięci lub jedną. Czy wiele jednostek, każda o innej architekturze i innym cyklu działania będzie poprawnie współpracować przy zwiększaniu skali rozwiązania?
3. Czy wraz ze wzrostem liczby rdzeni pamięć współdzielona nie stanie się „wąskim gardłem” systemu?
4. System RTOS zapewnia synchronizację zadań i wymianę danych w obrębie zadania z określonym, stałym odstępem czasu. W proponowanej architekturze wymiana danych odbywa

się natomiast wyłącznie na początku i na końcu cyklu sterowania. Jakie formalne przesłanki pozwalają uznać taki model za wystarczający dla układów o silnie asynchronicznej dynamice zdarzeń?

5. Jakie własności deterministyczne środowiska wykonawczego zostały uzyskane dzięki rezygnacji z RTOS i rozbiciu zadań na osobne, asynchronicznie działające rdzenie, a jakie zostały utracone względem systemów klasycznych?
6. Wydaje się, że dalszy rozwój oprogramowania CPDev powinien uwzględniać modele denotacyjne. Ciekawym problemem może być np. badanie odporności wielordzeniowego systemu zawierającego moduły transmisji (sieciowej i WiFi), zbudowanego z zastosowaniem modelu denotacyjnego. Czy takie próby były prowadzone?

5. Uwagi natury redakcyjnej

Praca jest zredagowana starannie, zawiera dużą liczbę ilustracji (w większości kolorowych), co bardzo ułatwia analizę otrzymanych rezultatów pochodzących z badań laboratoryjnych. Ten proces wspomagają również tabele, które pozwalają na wnikanie w szczegóły wyników. Wydaje się, że praca nie zawiera błędów edytorskich. Jest to prawdopodobnie wynikiem uważnej korekty autorskiej tekstu.

6. Podsumowanie

Podsumowując stwierdzam, że Doktorant z sukcesem rozwiązał problem badawczy uzyskując istotny i funkcjonalny wynik. Cel pracy został zrealizowany poprzez opracowanie i wdrożenie koncepcji środowiska wielordzeniowego rozszerzającego możliwości opracowanego wcześniej w Katedrze Informatyki i Automatyki Politechniki Rzeszowskiej jednordzeniowego środowiska CPDev.

Zaproponowaną metodę projektowania i tworzenia maszyny wirtualnej dla sterowników programowalnych, można uznać za oryginalny wkład Doktoranta do dyscypliny naukowej *Informatyka Techniczna i Telekomunikacja*. Rozwiązane zagadnienia mogą być wykorzystane praktycznie, co potwierdzają wyniki badań przeprowadzonych dla dwóch przykładowych aplikacji.

Doktorant wykazał się umiejętnością poprawnego wyboru i sformułowania naukowego celu pracy, do którego konsekwentnie zdążył i cel ten osiągnął. Jest to dowód na to, że posiada umiejętność prowadzenia samodzielnej pracy naukowej. Ponadto widoczna jest duża sprawność Doktoranta jako programisty nisko- i wysokopoziomowego.

Na podstawie bazy SCOPUS można stwierdzić, że Pan mgr inż. Marcin Hubacz ma w swoim dorobku współautorstwo 8 artykułów z listy MNiSW, w takich czasopismach, jak: *Electronics Swicerland* (2 prace), *Archives of Control Sciences* (1 praca), *Periodica Polytechnica Electrical Engineering and Computer Science* (1 praca), oraz 4 artykuły w materiałach konferencyjnych (w 5 artykułach jest pierwszym autorem). Dorobek ten spełnia z nadmiarem wymagania stawiane pracom doktorskim.

7. Wniosek końcowy

Podsumowując stwierdzam, że wymienione wyżej osiągnięcia stanowią wkład Doktoranta w rozwój dyscypliny *Informatyka techniczna i telekomunikacja*. Uważam, że rozprawa doktorska mgr. inż. Marcina Hurbacza spełnia wymagania stawiane rozprawom doktorskim przez obowiązującą Ustawę z dnia 20 lipca 2018r. *Prawo o szkolnictwie wyższym i nauce* (Dz.U. z 2021r. poz.478 z późn. zm.) i wnioskuję o jej dopuszczenie do publicznej obrony.

A handwritten signature in black ink, appearing to be the initials 'JH', is located in the lower right quadrant of the page.