

Analiza mocy obliczeniowej platform sprzętowych dla wieloosiowego sterownika napędów bezpośrednich

Grzegorz Góra, Piotr Mars, Maciej Petko, Grzegorz Karpel

1. Wstęp

Klasyczna architektura układu sterowania urządzeń mechatronicznych składa się z jednego nadrzędnego sterownika (ang. *Main Controller*) oraz sterowników osi (ang. *Joint Controller*), skonfigurowanych do pracy w trybie pozycyjnym, prędkościowym lub momentowym. Wynika to z podziału funkcjonalnego systemu oraz rozłożenia mocy obliczeniowej i zasobów sprzętowych na kilka niezależnych jednostek. Jednak współczesne platformy sprzętowe, dysponujące wysokimi mocami obliczeniowymi oraz posiadające dużą ilość zasobów sprzętowych w postaci interfejsów, wbudowanych modułów oraz portów ogólnego przeznaczenia, pozwalają na integrację sterownika głównego oraz sterowników osi w jednym układzie scalonym. Integracja sterownika nadrzędnego oraz sterowników osi pozwala:

- zwiększyć niezawodność poprzez wyeliminowanie zewnętrznych magistral wymiany danych pomiędzy sterownikiem głównym a sterownikami napędów;
- poprawić dokładność sterowania poprzez wyeliminowanie zewnętrznych torów analogowych służących do zadawania prędkości i momentu, które są podatne na szumy i zakłócenia oraz wprowadzają dodatkowe opóźnienie;
- obniżyć koszty poprzez zastąpienie kilku układów scalonych jedną platformą sprzętową;
- wyeliminować wewnętrzne interfejsy wymiany danych;
- wyeliminować konieczność korzystania z dwóch enkoderów (z których jeden dostarcza informację o położeniu dla sterownika nadrzędnego, a drugi dla układu komutacji elektronicznej) lub enkodera o podwójnym interfejsie.

Streszczenie: Klasyczna architektura układu sterowania urządzeń mechatronicznych składa się z jednego nadrzędnego sterownika oraz sterowników osi, skonfigurowanych do pracy w trybie pozycyjnym, prędkościowym lub momentowym. Wynika to z podziału funkcjonalnego systemu oraz rozłożenia mocy obliczeniowej i zasobów sprzętowych na kilka niezależnych jednostek. Jednak współczesne platformy sprzętowe, dysponujące wysokimi mocami obliczeniowymi oraz posiadające dużą ilość zasobów, w postaci sprzętowych interfejsów, wbudowanych modułów oraz portów ogólnego przeznaczenia, pozwalają na integrację sterownika głównego oraz sterowników osi w jednym układzie scalonym.

W artykule przedstawiono porównanie mocy obliczeniowej siedmiu wer-

sji systemów bazujących na układach FPGA oraz mikrokontrolerach z rdzeniem ARM-Cortex Mx. Testów wydajności dokonano poprzez implementację pętli prądowej sterownika napędu bezpośredniego, składającej się z transformacji Clarke i Parka, regulatora PI, modułu normalizacji jednostek oraz modulatora typu SPWM. Przedstawiono również poziom wykorzystania zasobów sprzętowych układu FPGA w przypadku użycia softprocesora Nios II, wspomaganego sprzętową jednostką zmiennoprzecinkową pojedynczej precyzji FPU oraz dodatkowymi instrukcjami koprocesora do obliczenia funkcji trygonometrycznych.

Słowa kluczowe: moc obliczeniowa, mikrokontrolery ARM Cortex, układ FPGA, sterownik prądowy

🇬🇧 ANALYSIS OF COMPUTING POWER OF HARDWARE PLATFORMS FOR A MULTI-AXIS CONTROLLER OF DIRECT DRIVES

Abstract: The classic control system architecture of mechatronic devices consists of one master controller and axle controllers configured to work in positional, velocity or torque mode. This is due to the functional system division and the distribution of computing power and hardware resources to several independent units. However, modern hardware platforms with high computing power having a large amount of resources, in the form of hardware interfaces, built-in modules and general purpose ports enable integration of the main controller and axis controllers into a single integrated circuit.

The article presents a comparison of the computing power of seven versions

of systems based on FPGA chips and microcontrollers with ARM-Cortex Mx core. Performance tests were carried out by the direct drive controller's current loop implementation consisting of the Clarke and Park transforms, the PI controller, the unit normalization module and the SPWM type modulator. The level of the FPGA system hardware resources utilization was also presented in the case of Nios II soft processor usage, supported by the single-precision floating-point FPU hardware unit and additional coprocessor instructions for the trigonometric functions calculation.

Keywords: computing power, ARM Cortex microcontrollers, FPGA, current loop

Integracja całego układu sterowania w jednym układzie scalonym jest rozwiązaniem przeznaczonym przede wszystkim dla aplikacji wykorzystujących napędy małej i średniej mocy. W tego typu rozwiązaniach, jak: platformy jezdne, głowice obserwacyjne czy mini-manipulatory, można zaimplementować sterownik całego urządzenia mechatronicznego w postaci jednego obwodu drukowanego PCB (ang. *Printed Circuit Board*), oszczędzając wykorzystywaną przestrzeń i obniżając koszty produktu.

2. Przegląd platform sprzętowych

Platforma sprzętowa przeznaczona do implementacji wieloosiowego sterownika napędów bezpośrednich musi łączyć w sobie zarówno wysoką wydajność obliczeniową, jak i elastyczną architekturę, umożliwiającą implementację wielu interfejsów do komunikacji i odczytu danych z urządzeń peryferyjnych, takich jak: enkodery, czujniki Halla, przetworniki ADC (ang. *Analog to Digital Converter*). Dodatkowo od platformy sprzętowej wymaga się zachowania stałej częstotliwości próbkowania dla wszystkich kanałów (osi) sterownika oraz możliwość wykorzystania dużej liczby portów wejściowych i wyjściowych. Najczęściej w aplikacjach o takich wymaganiach wykorzystuje się mikrokontrolery ogólnego przeznaczenia, mikrokontrolery przeznaczone do cyfrowego przetwarzania sygnałów DSP (ang. *Digital Signal Processing*), a także układy FPGA (ang. *Field Programmable Gate Array*).

2.1. Mikrokontrolery z rdzeniem ARM

Mikroprocesory z rdzeniem ARM Cortex występują w trzech wersjach.

Pierwszą grupą są procesory aplikacyjne (Cortex-Ax). Zaawansowane 32-bitowe i 64-bitowe procesory przeznaczone do wykorzystania w aplikacjach wymagających dużej mocy obliczeniowej. Najczęściej pracują pod kontrolą systemów operacyjnych, takich jak Android, Linux lub Windows CE. Posiadają wysoką częstotliwość taktowania rdzenia (powyżej 1 GHz), moduł zarządzania pamięcią MMU (ang. *Memory Management Unit*), rozszerzenia umożliwiające wirtualizację (np. implementację

wirtualnej maszyny Javy). Wykorzystywane są najczęściej w komputerach przenośnych, telefonach komórkowych oraz tabletach [2, 3].

Kolejną grupą są procesory dedykowane do aplikacji tzw. czasu rzeczywistego (Cortex-Rx). 32-bitowe procesory przeznaczone do zadań, w których krytyczny jest czas wykonywania pewnych operacji. Oferują szereg funkcji wspomagających implementację aplikacji bezpieczeństwa online. Większość z tych procesorów nie posiada MMU tylko podstawowy moduł ochrony pamięci MPU (ang. *Memory Protection Unit*). Taktowane są zegarem o częstotliwości kilkuset MHz (od 200 MHz do nieco powyżej 1 GHz). Umożliwiają wykorzystanie okrojonych wersji systemów operacyjnych (np. Linux) lub tzw. systemów operacyjnych czasu rzeczywistego RTOS (ang. *Real Time Operating System*). Wykorzystywane są najczęściej w aplikacjach samochodowych (np. ABS, układ napędowy), kontrolerach dysku twardego oraz komunikacji bezprzewodowej [4, 5]. Podobnie jak wersja Ax, mimo dużej wydajności obliczeniowej nie dysponują (lub dysponują w niewielkim stopniu) sprzętowymi, wbudowanymi interfejsami do obsługi urządzeń peryferyjnych.

Ostatnią grupą są procesory wykorzystywane jako mikrokontrolery (Cortex-Mx). 32-bitowe procesory, posiadające zintegrowaną pamięć oraz dużą liczbę układów peryferyjnych. Seria ta jest zoptymalizowana pod kątem niskiego zużycia energii oraz niskiej ceny [10]. Taktowane są zegarem o częstotliwości od kilkunastu do kilkuset megaherców. Stosowane są głównie jako mikrokontrolery ogólnego przeznaczenia, wykorzystywane w systemach sterowania, automatyki oraz w aplikacjach typu embedded [5].

2.2. Układy FPGA

W zaawansowanych systemach elektronicznych, w których krytyczny jest czas wykonania operacji lub wymagana jest implementacja algorytmów o dużej złożoności obliczeniowej, powszechnie wykorzystuje się układy rekonfigurowalne. Pozwalają one na równoległe wykonywanie zadań, umożliwiają implementację dowolnego modułu w postaci sprzętowej oraz zapewniają dużą

elastyczność w doborze dedykowanej konfiguracji pomiędzy sprzętem i oprogramowaniem, co jest niezwykle cennym atutem, zwłaszcza w połączeniu z możliwością wielokrotnego zmieniania dobranej konfiguracji [6]. Dodatkowymi zaletami układów typu FPGA jest duża liczba dostępnych portów ogólnego przeznaczenia oraz możliwość zachowania stałej częstotliwości próbkowania poprzez podział systemu na niezależne, równoległe wykonywane zadania.

Bezpośrednia implementacja algorytmów przy pomocy języków opisu sprzętu HDL (ang. *Hardware Description Language*) jest tylko jedną z możliwości wykorzystania układu FPGA jako platformy sprzętowej. Dodatkowe możliwości zapewniają systemy softprocesorowe, które są funkcjonalnie odpowiednikiem mikrokontrolera zaimplementowanego w programowalnej logice układu FPGA. W przeciwieństwie do standardowego mikroprocesora nie posiadają stałej architektury zbudowanej na krzemowym podłożu, lecz są rodzajem złożonej maszyny stanu opisaną w językach opisu sprzętu HDL. Podstawowym elementem systemu softprocesorowego firmy Altera jest 32-bitowy rdzeń Nios II o architekturze typu RISC (ang. *Reduced Instruction Set Computing*), pozwalający na implementację mikrokontrolera ogólnego przeznaczenia. Dodatkowo struktura jest uzupełniona o pamięć, interfejsy do jej obsługi oraz zestaw układów peryferyjnych [7]. Podobnie jak inne rodziny mikrokontrolerów, wszystkie procesory oparte na rdzeniu Nios II wykorzystują spójny zestaw instrukcji i model programowania. Producent dostarcza w pełni funkcjonalny zestaw narzędzi, służący do programowania procesora w językach wysokiego poziomu C/C++ oraz do jego debugowania i testowania. Do tego producent dostarcza zestaw standardowych układów peryferyjnych powszechnie używanych w mikrokontrolerach, takich jak: liczniki, interfejsy komunikacji szeregowej, kontrolery SDRAM oraz innych pamięci. Dodatkowo użytkownik może zaimplementować własne komponenty, zarówno jako układy peryferyjne przeznaczone do komunikacji z otoczeniem, jak i bloki sprzętowe dedykowane do wykonania odpowiednich zadań [7].

Jedną z charakterystycznych cech procesorów Nios II jest możliwość rozszerzenia listy instrukcji o instrukcje własne projektanta (ang. *Custom Instructions*). Oznacza to, że projektant może dołączyć do rdzenia Nios II blok logiczny rozszerzający możliwości wbudowanej jednostki arytmetyczno-logicznej (np. o funkcje trygonometryczne) [7].

3. Testy wydajności obliczeniowej

3.1. Sterownik prądowy

Aby przetestować wydajność obliczeniową platform sprzętowych, zaimplementowano podstawową wersję sterownika w postaci pętli prądowej układu sterowania. Test ma na celu sprawdzenie i porównanie wydajności obliczeniowej różnych systemów bazujących na badanych platformach sprzętowych oraz określenie poziomu wykorzystania zasobów sprzętowych w przypadku implementacji w układzie FPGA. Celem testu jest wyznaczenie liczby sterowników prądowych o częstotliwości próbkowania 10 kHz, które można zaimplementować na: mikrokontrolerach z rdzeniem ARM Cortex Mx, softprocesorze Nios II wspomaganym dodatkowymi modułami sprzętowymi oraz w wyniku implementacji modułów przy pomocy języków opisu sprzętu bezpośrednio w układzie FPGA.

Jako dane wejściowe do testu przygotowano przebiegi czasowe funkcji: prądu zadanego, prądów fazowych z dwóch

faz oraz kąta położenia wału. Dane te miały symulować wartości odczytane z enkodera, układów pomiaru prądów fazowych oraz wartość zadaną. Wszystkie dane wejściowe były przygotowane w formacie stałoprzecinkowym, ponieważ zarówno wartość odczytana z enkodera, jak i wartości prądów otrzymane za pośrednictwem przetworników analogowo-cyfrowych są wartościami całkowitoliczbowymi. Również rezultaty obliczeń, czyli współczynniki wypełnienia PWM dla każdej z faz, przyjmują wartości całkowitoliczbowe, które są następnie podawane na modulator w postaci odpowiednio skonfigurowanego licznika (ang. *timer*) w przypadku mikrokontrolerów lub modułu sprzętowego w przypadku układu FPGA. Wszystkie magistrale wejściowe i wyjściowe posiadają szerokość 16-bitów. Czasy wykonywania operacji były mierzone przez zewnętrzny układ FPGA z rozdzielczością 10 ns, a rozpoczęcie i ukończenie procesu obliczeniowego sygnalizowane wystawieniem flagi na jeden z portów ogólnego przeznaczenia. Testowany sterownik składał się z funkcji lub bloków sprzętowych o następujących zadaniach:

- normalizacja – w przypadku procesorów rzutowanie z formatu całkowitoliczbowego na zmiennoprzecinkowy oraz normalizacja do jednostek podstawowych (amper, radian), w przypadku modułu FPGA

normalizacja z kąta mechanicznego do kąta elektrycznego;

- transformata Clarke – obliczenia transformaty zgodnie ze wzorami 1 i 2 [1];

$$I_{\infty} = I_U \quad (1)$$

$$I_{\beta} = \frac{1}{\sqrt{3}} I_U + \frac{2}{\sqrt{3}} I_V \quad (2)$$

- transformata Parka – obliczenia transformaty zgodnie ze wzorami 3 i 4 [1];

$$I_d = I_{\infty} \cos(\theta) + I_{\beta} \sin(\theta) \quad (3)$$

$$I_q = -I_{\infty} \sin(\theta) + I_{\beta} \cos(\theta) \quad (4)$$

- regulator – podwójny regulator typu PI dla każdej ze składowej d i q ;
- modulator SPWM – obliczenia współczynników wypełnień dla każdej z faz UVW na podstawie kąta położenia wału oraz parametrów wyznaczonych przez regulator.

3.2. Wersje systemu

Testy wydajności obliczeniowej poszczególnych platform przeprowadzono, przygotowując 7 różnych systemów przy użyciu procesorów z rdzeniami ARM Cortex-Mx oraz układu FPGA Cyclone IV EP4CE115F29C7 firmy Altera. Wszystkie mikrokontrolery z rdzeniem ARM pracowały z maksymalną częstotliwością taktowania

rdzenia. Jako platformę FPGA wykorzystano układ o zasobach: 114 480 podstawowych elementów logicznych (ang. *Total logic elements*), 3888 Kbits wbudowanej pamięci (ang. *Embedded memory*), 266 wbudowanych układów mnożących (ang. *Embedded 18 × 18 multipliers*), czasie propagacji przez element logiczny: 7 ns [11]. Przygotowano następujące wersje systemów:

- **Cortex M3** – system wykorzystujący mikrokontroler STM32F103VBT6 firmy ST z rdzeniem ARM Cortex-M3; maksymalna częstotliwość taktowania rdzenia 72 MHz; wydajność 1.25 DMIPS/MHz; pamięci: 128 kB (Flash) i 20 kB (RAM) [8];
- **Cortex M4** – system wykorzystujący mikrokontroler STM32F407VGT6 firmy ST z rdzeniem ARM Cortex-M4 z wbudowaną jednostką zmiennoprzecinkową pojedynczej precyzji FPU; maksymalna częstotliwość taktowania rdzenia 168 MHz; wydajność: 210 DMIPS/1.25 DMIPS/MHz; pamięci: 1 MB (Flash), 192 kB (RAM) [9];
- **Nios** – system wykorzystujący procesor Nios II zaimplementowany na układzie Cyclone IV; system taktowany był zegarem podstawowym o częstotliwości 50 MHz; wykorzystywał wbudowaną pamięć (ang. *On-chip memory*); bez wsparcia sprzętowego dla arytmetyki zmiennoprzecinkowej; wszystkie operacje zmiennoprzecinkowe były wykonywane programowo przy użyciu biblioteki matematycznej dostarczonej ze środowiskiem kompilatora;
- **Nios (FPU)** – system wykorzystujący procesor Nios II zaimplementowany na układzie Cyclone IV; system taktowany był zegarem podstawowym o częstotliwości 50 MHz; wykorzystywał wbudowaną pamięć (ang. *On-chip memory*); obliczenia arytmetyczne były wspomagane wykorzystaniem sprzętowej jednostki zmiennoprzecinkowej pojedynczej precyzji FPU, wspomagającej operacje: dodawanie, odejmowanie, mnożenie i dzielenie;
- **Nios (Cordic)** – system wykorzystujący procesor Nios II zaimplementowany na układzie Cyclone IV; system taktowany był zegarem podstawowym o częstotliwości 50 MHz; wykorzystywał wbudowaną pamięć (ang. *On-chip memory*); jednostka

Tabela 1. Porównanie czasów wykonywania obliczeń przez poszczególne systemy

Platforma sprzętowa	Czas wykonania obliczeń [us]	
	Czas średni	Czas maks.
Cortex M3	137,59	149,39
Cortex M4	14,98	15,86
Nios	724,78	909,29
Nios (FPU)	705,74	892,89
Nios (Cordic)	77,80	81,49
Nios (FPU, Cordic)	62,91	65,73
Hardware FPGA	6,24	6,24

arytmetyczno-logiczna wspomagana była zaimplementowanymi instrukcjami do sprzętowego obliczania funkcji trygonometrycznych (sin, cos) przy pomocy instrukcji użytkownika (ang. *Custom Instructions*);

- **Nios (FPU, Cordic)** – system wykorzystujący procesor Nios II zaimplementowany na układzie Cyclone IV; system taktowany był zegarem podstawowym o częstotliwości 50 MHz; wykorzystywał wbudowaną pamięć (ang. *On-chip memory*); jednostka arytmetyczno-logiczna wspomagana była zarówno przez jednostkę zmiennoprzecinkową pojedynczej precyzji FPU, jak i sprzętowymi implementacjami funkcji trygonometrycznych;
- **Hardware** – system zaimplementowany na układzie Cyclone IV; taktowany był zegarem podstawowym o częstotliwości 50 MHz; algorytm był w całości zaimplementowany w postaci bloków sprzętowych opisanych w języku VHDL; wszystkie operacje wykonywane były przy użyciu arytmetyki stałoprzecinkowej.

3.3. Wyniki testu

Wyniki zaprezentowane w tabeli 1 przedstawiają maksymalne i średnie czasy wykonania wszystkich wymaganych obliczeń arytmetycznych dla pętli prądowej sterownika.

W tabeli 2 przedstawione jest porównanie wykorzystania zasobów sprzętowych przez poszczególne wersje systemów zaimplementowanych na układzie FPGA. W nawiasach podano procentowy stopień wykorzystania zasobów sprzętowych układu.

Tabela 2. Wykorzystanie zasobów sprzętowych przez poszczególne systemy

Platforma sprzętowa	Elementy logiczne	Rejestry	Bity pamięci
Nios	2882 (3%)	1754 (2%)	325 504 (8%)
Nios (FPU)	6923 (6%)	1987 (2%)	325 504 (8%)
Nios (Cordic)	5590 (5%)	2102 (2%)	325 504 (8%)
Nios (FPU, Cordic)	9657 (8%)	2305 (2%)	325 504 (8%)
Hardware FPGA	3103 (3%)	2765 (2%)	0 (0%)

4. Wnioski

Zakładając częstotliwość próbkowania sterownika na poziomie 10 kHz (100 us), systemy bazujące na procesorze z rdzeniem Cortex M3, softprocesorze Nios II oraz soft-procesorze Nios II z jednostką FPU nie pozwalają na zaimplementowanie pętli prądowej sterownika z zakładaną częstotliwością próbkowania. Procesor Nios II wspomagany sprzętowymi instrukcjami trygonometrycznymi oraz procesor wspomagany funkcjami trygonometrycznymi i koprocesorem FPU pozwalają na zaimplementowanie jednego sterownika pracującego z zakładaną częstotliwością. Dzięki wykorzystaniu układu FPGA jako platformy sprzętowej istnieje możliwość zwiększenia ilości rdzeni tak, aby każdy softprocesor odpowiadał za jeden sterownik. Tak przygotowany system posiada jednak niewielką rezerwę mocy obliczeniowej, dlatego implementacja bardziej zaawansowanego algorytmu stopowania spowoduje konieczność obniżenia częstotliwości próbkowania. Procesor z rdzeniem Cortex-M4, który jest wyposażony we wbudowaną jednostkę FPU, pozwala na zaimplementowanie maksymalnie do 6 sterowników prądowych. Trzeba jednak zaznaczyć, że rdzeń był taktowany najwyższą dostępną częstotliwością, a testowany algorytm był najprostszą wersją pętli prądowej, bez modelu matematycznego napędu i z podstawowym regulatorem. Wykorzystanie bardziej złożonej metody sterowania (np. bazującej na modelu), dodanie pętli prędkościowej i/lub pozycyjnej, dodanie obsługi interfejsów komunikacyjnych lub zwiększenie częstotliwości

próbkowania spowoduje znaczny spadek możliwości implementacji sterowników na tej platformie.

Wykorzystanie układu FPGA poprzez implementację softprocesora dla każdego sterownika pociąga za sobą znaczne wykorzystanie zasobów sprzętowych układu zgodnie z wynikami przedstawionymi w tabeli 2. Jedynie zastosowanie układu FPGA poprzez implementację bloków sprzętowych pozwala na spełnienie założeń dotyczących wysokiej wydajności obliczeniowej przy zachowaniu niskiego wykorzystania zasobów sprzętowych. Dodatkową zaletą układów reprogramowalnych jest ułatwiona możliwość implementacji dowolnego niestandardowego interfejsu do komunikacji z układami peryferyjnymi.

Literatura

- [1] RUDNICKI T., CZERWIŃSKI R., FRĘCHOWICZ A.: *Układy sterowania silnikiem PMSM*. „Maszyny Elektryczne – Zeszyty Problemowe” 90/2011.
- [2] ARM: *ARM® Cortex®-A Series, Programmer's Guide for ARMv8-A*. <http://www.arm.com>, Version: 1.0, ARM DEN0024A, ID050815, 2015.
- [3] PAPROCKI K.: *Mikrokontrolery STM32 w praktyce*. Wydawnictwo BTC, Legonowo 2011.
- [4] CRASKE S.: *ARM Cortex-R Architecture, For Integrated Control and Safety Applications*. <http://www.arm.com>, 2013.
- [5] YIU J.: *ARM® Cortex®-M for Beginners, An overview of the ARM Cortex-M processor family and comparison*. <http://www.arm.com>, 2016.
- [6] KARPIEL G.: *Zastosowanie podejścia mechatronicznego w projektowaniu robotów równoległych*. Praca doktorska, Akademia Górniczo-Hutnicza im. Stanisława Staszica, Kraków 2006.
- [7] Altera Corporation: *Nios II Classic Processor Reference Guide*. <http://www.altera.com>, 2016.
- [8] STMicroelectronics: *STM32F103x8, STM32F103xB*. <http://www.st.com>, DocID13587 Rev 17, 2015.
- [9] STMicroelectronics: *STM32F405xx, STM32F407xx*. <http://www.st.com>, DocID022152 Rev 8, 2016.
- [10] SOTOR D., MICHALAK J.: *Mikroprocesorowy system prototypowania do testowania układów sterowania przekształtników*. „Maszyny Elektryczne – Zeszyty Problemowe” 2/2015.
- [11] <http://www.terasic.com.tw>

mgr inż. Grzegorz Góra,
mgr inż. Piotr Mars,
dr hab. inż. Maciej Petko,
dr inż. Grzegorz Karpiel
AGH Akademia Górniczo-Hutnicza,
Kraków

artykuł recenzowany