

Grzegorz Góra, Grzegorz Karpiel, Piotr Mars, Radosław Sitek, Marek Goczał  
AGH Akademia Górniczo-Hutnicza, Kraków; Energylandia, Zator

## SPRZĘTOWA IMPLEMENTACJA TRANSFORMACJI CLARKA I PARKA

### HARDWARE IMPLEMENTATION OF THE CLARKE AND PARK TRANSFORMATIONS

**Streszczenie:** Wykorzystanie układu FPGA jako platformy sprzętowej pozwala na równoległe przetwarzanie zadań oraz implementację wielu systemów na jednym układzie fizycznym. Te zalety układów reprogramowalnych skłoniły autorów do przedstawienia koncepcji systemu, który pozwala na implementację wielu sterowników dla napędów bezpośrednich małych i średnich mocy na jednym układzie fizycznym. Implementacja transformacji Clarka i Parka jest konieczna, aby proponowany system dawał możliwość zaprojektowania sterowników pracujących w trybie prądowym. Elementami zaprojektowanego modułu sprzętowej transformacji prądów fazowych do współrzędnych (d,q) są: komponent wykorzystujący algorytm CORDIC do wyznaczania wartości funkcji trygonometrycznych kąta wejściowego, zaimplementowany algorytm mnożenia sekwencyjnego oraz maszyna stanu, której zadaniem jest wyznaczenie transformaty przy pomocy wcześniej opisanych komponentów oraz podstawowych operacji arytmetycznych i logicznych. W celu weryfikacji zaimplementowanego modułu sprzętowego przygotowano system bazujący na soft-procesorze Nios II umożliwiający: przygotowanie pseudolosowych danych wejściowych, weryfikację poprawności otrzymywanych wyników oraz pomiar czasu wykonywania transformacji przez testowany moduł. W podsumowaniu omówiono otrzymane błędy numeryczne oraz przedstawiono zalety i wady opracowanego modułu.

**Abstract:** The FPGA chip utilization as a hardware platform enables the parallel processing of tasks and multiple systems on a single chip implementation. These advantages of the reprogrammable chips led authors to present the concept of the system, which allows the implementation of multiple controllers for the small and medium power direct drives on a single physical system. The Clarke and Park transformations implementation is essential to give the possibility to design the working in current mode controllers for the proposed system. The designed module elements for hardware transformation of phase currents to the (d,q) coordinates are: component utilizing CORDIC algorithm for the calculation of trigonometric functions values of the input angle, the implemented sequential multiplication algorithm as well as the state machine which aims to determine the transform using the previously described components and basic arithmetic and logic operations. In order to verify the implemented hardware module, the system based on soft-processor Nios II was prepared which enables: preparation of pseudo-random input data, verification of the correctness of the obtained results, and the transformation execution time measurement of the tested module. In the conclusion, the obtained numerical errors were discussed as well as advantages and disadvantages of the developed module was presented.

**Słowa kluczowe:** transformacja Clarka, Transformacja Parka, FPGA, CORDIC

**Keywords:** Clarke Transform, Park transform, FPGA, CORDIC

### 1. Wstęp

Układy FPGA (*ang. Field-Programmable Gate Array*) w ostatnich latach systematycznie zdobywają rynek aplikacji SoC (*ang. System-on-a-Chip*), co wynika z ich możliwości wykonywania równoległe zadań, tworzenia dedykowanych bloków i modułów sprzętowych, możliwości elastycznej budowy systemu oraz dużej liczby portów ogólnego przeznaczenia.[11] Układ FPGA to zintegrowany obwód stanowiący tablicę podstawowych bloków logicznych, w których użytkownik przy pomocy jednego z języków opisu sprzętu HDL (*ang.*

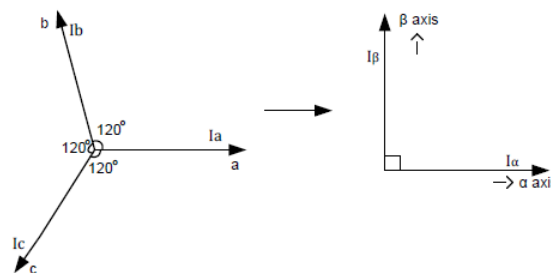
*Hardware Description Language*) może zdefiniować połączenia oraz zadać wewnętrzną konfigurację. Układy rekonfigurowalne FPGA łączą więc zalety procesorów ogólnego przeznaczenia oraz układów cyfrowych, przy czym szybkość działania i stopień wykorzystania układu zależą w głównej mierze od projektanta oraz zastosowanych algorytmów. Dzięki możliwości równoległego przetwarzania zadań oraz implementacji wielu systemów na jednym układzie fizycznym układ FPGA wybrano jako platformę sprzętową do zaprojektowania sys-

temu zintegrowanego sterownika dla wielu napędów bezpośrednich małych i średnich mocy. W proponowanym systemie użytkownik będzie miał możliwość zaprojektowania sterowania pozycyjnego, prędkościowego lub prądowego dla jednego lub kilku napędów typu PMSM (ang. *Permanent Magnet Synchronous Motor*) lub BLDC (ang. *BrushLess Direct Current*), korzystając z gotowych komponentów lub/i własnych modułów. Umieszczenie całego systemu sterowania w jednym układzie fizycznym pozwala obniżyć koszty, zwiększyć niezawodność poprzez wyeliminowanie zewnętrznych magistral wymiany danych pomiędzy układem nadrzędnym a sterownikami napędów oraz pomiędzy samymi sterownikami oraz poprawić jakość sterowania poprzez wyeliminowanie zewnętrznych torów analogowych służących do zadawania prędkości i przyspieszenia, które są podatne na szумы i zakłócenia, które są podatne na szумы i zakłócenia oraz wprowadzają dodatkowe opóźnienie. Podejście takie daje także możliwość uwzględnienia w modelu matematycznym systemu nadrzędnego większej liczby parametrów pracy napędów.

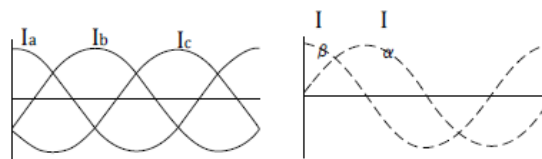
Projektowany system może być powszechnie wykorzystywany w aplikacjach typu SoC, jeśli zaimplementowane sterowanie dla jednego napędu nie zajmie więcej zasobów sprzętowych układu FPGA niż ok. 4 do 5 tys. elementów logicznych. Pozwoli to na zaprojektowanie systemu sterowania kilkoma napędami bezpośrednimi na powszechnie dostępnych i tanich układach o pojemności 20-30 tys. elementów logicznych i osiągnięcie zużycia zasobów logicznych porównywalnych z podstawowymi wersjami soft-procesorów (np. Nios II). Aby spełnić to założenie transformata Clarka i Parka, konieczna w przypadku implementacji sterowania prądowego nie może przekraczać 1 tys. elementów logicznych.

## 2. Transformacja Clarka i Parka

Transformata Clarka służy do odwzorowania wirującego w przestrzeni trójwymiarowej (a,b,c) wypadkowego wektora prądu na dwuosiowy stacjonarny układ współrzędnych prostokątnych ( $\alpha, \beta$ ) [2,3]. Oś „ $\alpha$ ” jest usytuowana zgodnie z wektorem fazy „a”, natomiast oś „ $\beta$ ” jest do niej prostopadła (Rys.1). Przebiegi wartości chwilowych prądów fazowych (a,b,c) oraz wektorów składowych  $\alpha, \beta$  w dziedzinie czasu przedstawiono na (Rys.2).



Rys. 1. Transformacja Clarka z przestrzeni trójwymiarowej a, b, c do przestrzeni dwuwymiarowej  $\alpha, \beta$  [2]



Rys. 2. Przebiegi prądów fazowych  $I_a, I_b, I_c$  oraz prądów składowych  $I_\alpha, I_\beta$  [2]

Wektor prądu jest przedstawiony za pomocą dwóch składowych  $I_\alpha$  oraz  $I_\beta$  zamiast trzech, które łatwo wyznaczyć za pomocą wzorów (1), (2), (3). Geometryczna suma prądów fazowych wynosi zawsze zero, stąd wystarczający jest tylko pomiar dwóch prądów fazowych, natomiast wartość trzeciego prądu fazowego wyznaczana jest na podstawie prostych obliczeń arytmetycznych.

$$I_\alpha = I_a \quad (1)$$

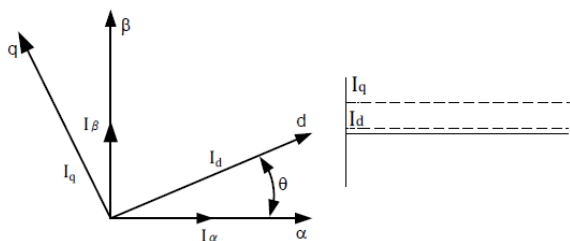
$$I_\beta = \frac{1}{\sqrt{3}} \cdot I_a + \frac{2}{\sqrt{3}} \cdot I_b \quad (2)$$

$$I_a + I_b + I_c = 0 \quad (3)$$

Transformacja Parka jako odniesienie przyjmuje wirujący układ współrzędnych prostokątnych (d,q) gdzie oś „d” jest zorientowana zgodnie z kierunkiem wirującego pola wirnika (Rys.3). Operacja przejścia z jednego układu współrzędnych do drugiego polega na rzutowaniu wartości składowych wektorów prądów  $I_\alpha, I_\beta$  na osie wirującego układu (d,q), co przedstawiają wzory: (4), (5). Parametr  $\theta$  występujący we wzorach na transformację, jest chwilowym położeniem kątowym wirnika [2,3].

$$I_d = I_\alpha \cdot \cos(\theta) + I_\beta \cdot \sin(\theta) \quad (4)$$

$$I_q = -I_\alpha \cdot \sin(\theta) + I_\beta \cdot \cos(\theta) \quad (5)$$



Rys. 3. Transformacja Parka ze stacjonarnej ramki  $\alpha$ ,  $\beta$  do wirującego układu współrzędnych  $d$ ,  $q$ . Przebiegi czasowe prądów  $I_d$   $I_q$  [2]

Dzięki zastosowaniu tej transformaty otrzymujemy dwie stałoprądowe składowe  $I_d$  oraz  $I_q$ , które posiadają swoją interpretację fizyczną. Prąd  $I_d$  odpowiada wartości strumienia magnetycznego, natomiast  $I_q$  jest interpretowany jako wartość momentu napędowego silnika [4]. W celu osiągnięcia największej sprawności silnika, algorytm regulacji powinien minimalizować wartość składowej  $I_d$ , a maksymalizować wartość składowej  $I_q$  [6].

### 3. Sprzętowa implementacja transformacji Clarka i Parka

Projektowany sterownik powinien pracować z częstotliwością próbkowania maksymalnie do 20kHz (dla standardowego zegara 50MHz to 2500 cykli) obliczenia transformacji nie powinny przekraczać ok 20% czasu okresu próbkowania czyli 500 cykli zegara.

W związku z tym, że pomiar prądu w fazach silnika najczęściej odbywa się przez przetwornik analogowo-cyfrowy ADC (*ang. Analog to Digital Converter*) mierzący napięcie z czujnika prądu lub rezystora pomiarowego [1], zdecydowano że dane wejściowe będą przyjmowane w postaci podwójnej 16-bitowej magistrali z wartościami proporcjonalnymi do prądów w dwóch fazach silnika reprezentowanych w postaci całkowitoliczbowej ze znakiem w kodzie uzupełnień do dwóch U2. Pozwoli to w wielu przypadkach na bezpośrednie podłączenie wyjścia danych z modułu obsługi przetwornika ADC do bloku sprzętowego transformacji Clarka i Parka. Przy założeniu projektowym budowy sterownika małych i średnich mocy z prądem chwilowym maksymalnym  $\pm 25$  A, rozdzielczość pomiaru będzie wynosiła w najgorszym przypadku ok. 0.75 mA ( $2^{16} \sim 50$  A). W przypadku sterownika napędów bezpośrednich, rozdzielczość taka jest wystarczająca do prawidłowej pracy systemu, ponieważ czułość czujników do pomiaru prądu,

histereza charakterystyki napędów są około rząd wielkości większe. Wartości prądów we współrzędnych (d,q), po transformacji przedstawiane będą również w postaci dwóch 16-bitowych magistral w reprezentacji U2.

#### 3.1. Obliczanie funkcji trygonometrycznych

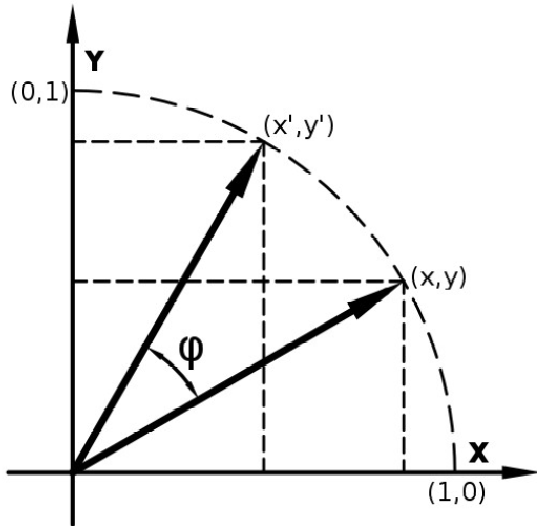
Obliczanie wartości funkcji trygonometrycznych w aplikacjach typu embeded działających w trybie on-line jest zadaniem wymagającym wykorzystania znacznych zasobów sprzętowych (elementy logiczne, pamięć) lub czasu obliczeń. W klasycznych zastosowaniach, gdy wartości te mogą być wyznaczane z niską dokładnością (do kilkunastu bitów) stosuje się tablicowanie wartości LUT (*ang. Look Up Table*) czyli, wykorzystanie struktury w formie tablicy do przechowywania wcześniej obliczonych wyników, co daje oszczędność czasu wymaganego na ich wyznaczenie kosztem większego zużycia pamięci. Niestety rozmiar tablicy szybko rośnie i ta metoda staje się mało efektywna dla wysokich rozdzielczości.

Druga metoda wykorzystuje aproksymację funkcji trygonometrycznych za pomocą wielomianów wysokich rzędów, co pozwala uzyskać duże dokładności otrzymywanych wyników w stosunku do metody tablicowania. Zwiększenie precyzji obliczeń odbywa się jednak kosztem wykonywania wielu podstawowych operacji arytmetycznych jak dodawanie, odejmowanie mnożenie lub/i dzielenie, dlatego też nie jest to rozwiązanie optymalne w przypadku implementacji na platformach sprzętowych, w których operacje mnożenia/dzielenia są czasochłonne.

Ostatnim podejściem możliwym do zastosowania, gdy istnieje możliwość ingerencji w konfigurację architektury sprzętu jest użycie dedykowanego algorytmu do obliczania funkcji hiperbolicznych i trygonometrycznych zwanego CORDIC (*ang. COordinate Rotation DIgital Computer*), który swoje działanie opiera jedynie na operacjach dodawania, odejmowania, przesunięciach bitowych oraz tablicy o rozmiarach najczęściej nieprzekraczających kilkudziesięciu pozycji. Brak operacji mnożeń oraz dzielen jest niewątpliwą zaletą tej metody.

Algorytm CORDIC opiera się na cyklicznym wyznaczaniu współrzędnych punktu znajdującego się na kole jednostkowym poprzez rotację aktualnej pozycji o pewien ustalony kąt [7]. Wartość tego kąta musi spełniać zależność:  $\tan(\varphi) = 2^{-1}$  dla  $i$  należącego do zbioru liczb

naturalnych, dlatego też jego wartość nie jest obliczana, w każdym kroku iteracji, lecz tablicowana [8]. Dzięki takiemu założeniu operacje mnożenia przez tangens kąta obrotu można zastąpić przesunięciem bitowym.



Rys. 4. Transformacja układów współrzędnych na kole jednostkowym

$$x' = x \cos(\varphi_i) - y \sin(\varphi_i) \quad (5)$$

$$y' = y \cos(\varphi_i) + x \sin(\varphi_i) \quad (6)$$

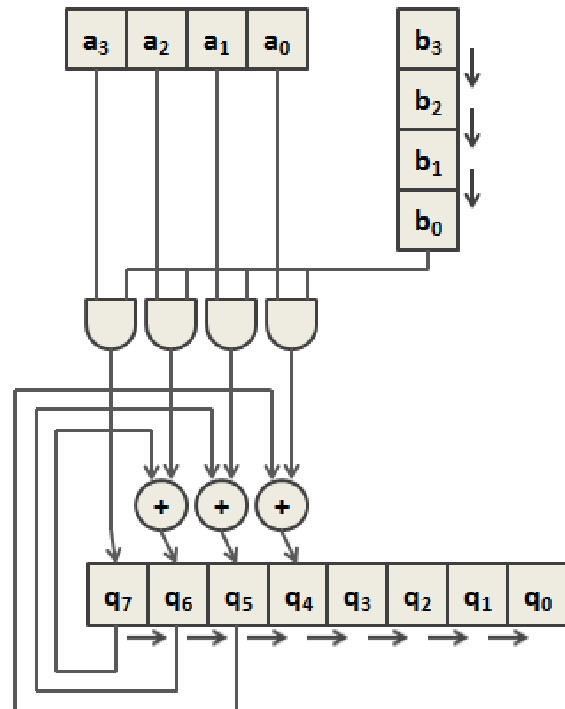
$$x' = \cos(\varphi_i) [x - y \operatorname{tg}(\varphi_i)] \quad (7)$$

$$y' = \cos(\varphi_i) [y + x \operatorname{tg}(\varphi_i)] \quad (8)$$

Składniki  $\cos(\varphi_i)$  ulegają skumulowaniu w postaci iloczynu, którego wartość jest stałą dla zadanej ilości kroków iteracji (dalej oznaczana jako C). Wartość stałej C jest zawsze dodatnia i nie zależy od znaku kąta obrotu. Dodatkowo końcowe mnożenie przez skumulowany współczynnik C można zastąpić ustawiając współrzędne początkowe punktu na  $[X_0, Y_0] = [C, 0]$ . Transformacja układów współrzędnych na kole jednostkowym (Rys. 4) oraz zależności ją opisujące (5-8) przedstawione są powyżej.

### 3.2. Mnożenie sekwencyjne

Projektowanie układu mnożącego na platformie FPGA zawsze jest kompromisem pomiędzy szybkością działania bloku sprzętowego a zajętością zasobów logicznych układu [9]. Najczęściej można wyróżnić dwa podstawowe algorytmy: mnożenie równoległe – stosowane gdy kluczowe są parametry częstotliwościowe i należy optymalizować układ mnożący pod kątem skrócenia czasu obliczeń oraz mnożenie sekwencyjne. Mnożenie sekwencyjne pozwala zaoszczędzić zasoby sprzętowe układu FPGA kosztem wydłużenia czasu otrzymania wyniku.



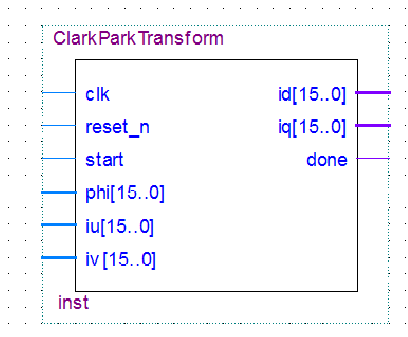
Rys. 5. Schemat blokowy 4-bitowego mnożenia sekwencyjnego

W przypadku implementacji bloku sprzętowego transformacji Clarka i Parka czas obliczeń transformaty będzie spełniał założenia systemu nawet w przypadku, gdy w każdym taktie zegara wyznaczony zostanie jeden bit iloczynu. Układ mnożący będzie więc optymalizowany pod kątem zużycia zasobów logicznych, dlatego zdecydowano użyć algorytmu przesun i dodaj (*ang. shift and add*) [9,10], z powodu efektywnego wykorzystania zasobów sprzętowych. Danymi wejściowymi w mnożeniu sekwencyjnym są dwie 16-bitowe liczby ze znakiem przedstawione w reprezentacji U2, przy czym zakłada się, że argument a jest liczbą całkowitą (0-bitów po przecinku), natomiast argument b liczbą stałoprzecinkową z 15 bitami po przecinku. Wynikiem mnożenia jest liczba 16-bitowa w reprezentacji U2 znormalizowana do wartości całkowitoliczbowej (0-bitów po przecinku).

W tej metodzie iloczyn logiczny argumentu, a (mnożnika) oraz jednego bitu argumentu b (mnożnej) stanowi częściowy element wyniku końcowego. W każdym kroku iteracji otrzymany rezultat cząstkowy należy dodać do otrzymanych rezultatów z poprzednich kroków iteracji, a następnie przesunąć o jeden bit w prawo. Po ilości cykli wynikających z długości mnożonych wektorów otrzymany zostanie wynik końcowy. Na rys. 5 przedstawiony został

schemat blokowy mnożenia sekwencyjnego wykorzystującego algorytm shift-and-add dla czterobitowych wartości wejściowych.

### 3.3. Sprzętowy moduł transformacji Clarka i Parka

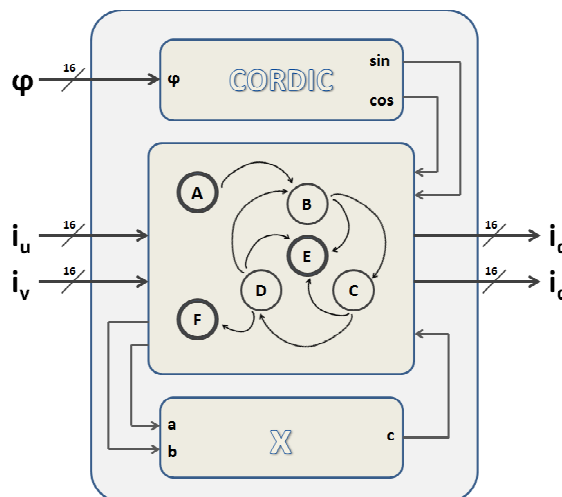


Rys. 6. Interfejs bloku sprzętowego transformacji Clarka i Parka

Blok do transformacji prądów fazowych do współrzędnych w wirującym układzie (d,q) związanym z wirnikiem składa się z trzech komponentów:

- zaimplementowanego algorytmu CORDIC do wyznaczenia wartości funkcji  $\sin(x)$  oraz  $\cos(x)$  kąta wejściowego,
- zaimplementowanego algorytmu mnożenia sekwencyjnego pozwalającego na otrzymanie iloczynu dwóch argumentów w kodzie uzupełnień do dwóch,
- prostej maszyny stanu, której zadaniem jest pobranie wartości wejściowych (prądów fazowych), wyznaczenie transformaty przy pomocy komponentów: CORDIC, mnożenia sekwencyjnego, oraz podstawowych operacji arytmetycznych i logicznych oraz zwrócenie wyniku końcowego.

W celu zapewnienia uniwersalności w zastosowaniach modułu w różnych rodzinach układów reprogramowalnych do jego implementacji wykorzystywano tylko podstawowe elementy logiczne, bez użycia wbudowanych mnożarek, bloków pamięci, układów pll, itd. Cały moduł został zoptymalizowany pod kątem niskiego wykorzystania zasobów sprzętowych, dzięki zastosowaniu odpowiednich algorytmów obliczeniowych. Schemat blokowy przedstawia rysunek 7.



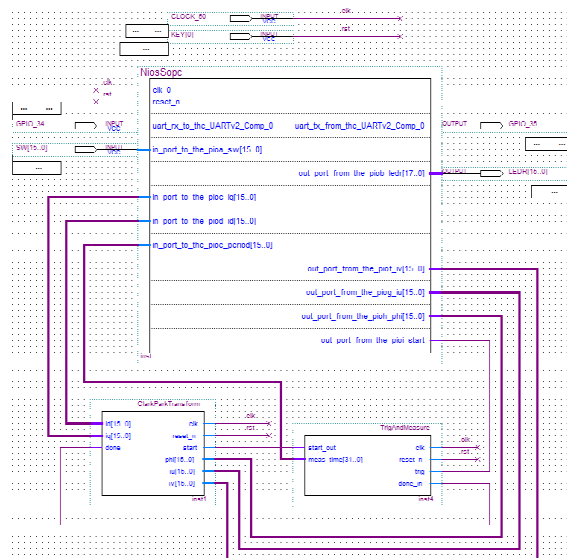
Rys. 7. Schemat blokowy modułu sprzętowej transformacji Clarka i Parka

## 4. Testy bloku sprzętowego

### 4.1. System do testowania i weryfikacji

Jako platformę sprzętową do implementacji oraz przeprowadzenia testów modułu transformacji prądów fazowych do prądów w wirującym układzie współrzędnych zastosowano układ Cyclone IV EP4CE115 firmy Altera w postaci zestawu edukacyjno-rozwojowego DE2-115 [12]. Płyta wyposażona jest w dużą liczbę układów peryferyjnych, co pozwala na implementację, testowanie oraz transmisję przez jeden z interfejsów szeregowych.

Do przeprowadzenia testów bloku sprzętowego ClarkParkTransform użyto soft-processor



Rys. 8. System do testowania bloku transformacji Clarka i Parka



Nios II, wyposażony dodatkowo w moduł UART (ang. *Universal Asynchronous Receiver and Transmitter*) do przesyłania wyników testu na komputer klasy PC (ang. *Personal Computer*) oraz pomocniczy moduł służący do pomiaru czasu wykonywania obliczeń przez weryfikowany moduł. Nios II jest 32-bitowym mikrokontrolerem typu RISC (ang. *Reduced Instruction Set Computer*) programowanym w języku C/C++, co pozwala na łatwe przygotowanie pseudolosowych danych wejściowych oraz porównanie wyników obliczeń na liczbach zmiennoprzecinkowych pojedynczej precyzji z rezultatami otrzymanymi z bloku ClarkParkTransform. Dane wejściowe, obliczone rezultaty oraz czasy obliczeń testowanego modułu wysyłane były przez interfejs UART to komputera. Tam, dokonano analizy błędów numerycznych oraz prezentacji rezultatów przeprowadzonych testów. Schemat blokowy systemu do weryfikacji modułu transformacji z programu Quartus II przedstawiony jest na rys. 8.

#### 4.2. Wyniki testów

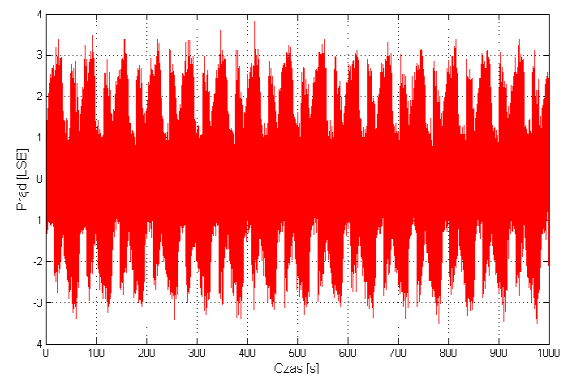
W celu weryfikacji poprawności wyników otrzymywanych przez zaimplementowany moduł sprzętowy wykorzystano wcześniej przygotowany system. Testy przeprowadzono dla miliona zestawów parametrów wejściowych (1000 s z próbkowaniem 1 kHz), którymi były pseudolosowe wartości prądów fazowych oraz kąta obrotu. Czas obliczeń transformaty był stały i niezależnie od wejściowych parametrów wynosił 141 cykli zegara taktującego moduł. Wykorzystanie zasobów sprzętowych układu FPGA dla całego bloku transformacji oraz dwóch jego podstawowych komponentów: wyznaczania wartości funkcji trygonometrycznych (CordicComp) i mnożenia sekwencyjnego (SeqMulComp) przedstawia tabela 1.

Tabela 1. Wykorzystanie zasobów sprzętowych przez moduł transformacji oraz poszczególne komponenty

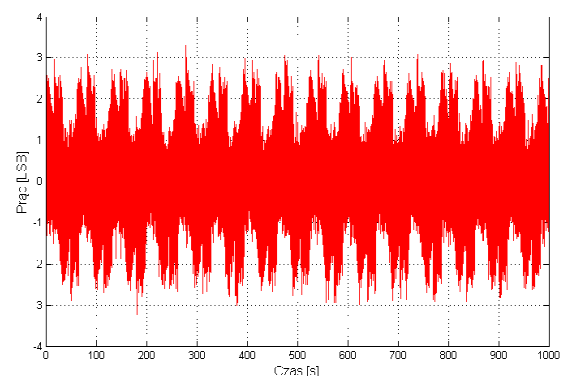
Blok sprzętowy	Elementy logiczne	Rejestry	Bitów pamięci	Mnożarki 9-bitowe
ClarkPark-Transform	805	434	0	0
CordicComp	443	116	0	0
SeqMulComp	152	128	0	0

Na rysunkach 9 oraz 10 przedstawione są przebiegi błędów numerycznych w porównaniu z obliczeniami na liczbach zmiennoprzecinko-

wych pojedynczej precyzji. Maksymalny błąd wyniósł 3.81 i 3.31, odpowiednio dla współrzędnej  $d$  i  $q$ , co przekłada się na błędy poniżej 3 mA dla największego zakładanego zakresu pomiaru prądu  $\pm 25$ A. Błąd średniokwadratowy MSE (ang. *Mean Squared Error*) dla obu składowych wyniósł około 0.5 mA. Dokładne wartości błędów przedstawione są w tabeli 2.



Rys. 9. Błąd numeryczny obliczeń: składowa  $d$  prądu



Rys. 10. Błąd numeryczny obliczeń: składowa  $q$  prądu

Tabela 2. Wartości błędów numerycznych

Wielkość	Błąd bezwzględny		Błąd przy maks. zakresie $\pm 25$ A	
	$d$ [LSB]	$q$ [LSB]	$d$ [mA]	$q$ [mA]
Błąd maksymalny	3.81	3.31	2.91	2.52
Błąd średniokwadratowy	0.707	0.598	0.54	0.46
Wartość średnia z funkcji błędu	0.002721	0.000764	0.002076	0.000583

#### 5. Wnioski

Przedstawiony w artykule moduł sprzętowej implementacji transformacji Clarka i Parka wykonujący operacje na 16-bitowych danych wejściowych pozwala na uzyskanie dokładności obliczeń z błędem maksymalnym nie przekraczającym 3 mA oraz błędem średnim około 0.5 mA dla największego zakresu pomiarowego

( $\pm 25$  A). Uzyskane dokładności pozwalają na użycie modułu jako elementu sterownika prądowego dla napędów bezpośrednich, ponieważ uzyskany błąd jest mniejszy od nieliniowości charakterystyk oraz poziomu szumów czujników do pomiaru prądu z badanego zakresu pomiarowego.

Wykorzystanie zasobów sprzętowych układu FPGA poniżej jednego tysiąca elementów logicznych pozwala na zastosowanie modułu do zaimplementowania sterowania wieloma napędami bezpośrednimi na jednym układzie fizycznym. Szukając metod dalszej optymalizacji pod kątem zajętości zasobów sprzętowych należy podjąć próbę optymalizacji komponentu do wyznaczania wartości funkcji trygonometrycznych, jako zajmującego 55% zasobów całego bloku sprzętowego. Dodatkowo wykorzystanie w projektowaniu tylko podstawowych elementów logicznych pozwala na implementację modułu w dowolnej rodzinie układów reprogramowalnych.

Krótki czas obliczeń transformaty Clarka i Parka w stosunku do okresu próbkowania sterownika (5.6% dla zegara taktującego 50 MHz i częstotliwości pracy sterownika 20 kHz) pozwala użyć modułu do obliczeń transformacji dla wielu napędów pod warunkiem odpowiedniego multipleksowania danych wejściowych i wyników.

## 6. Literatura

- [1]. T. Rudnicki, R. Czerwiński: "Pomiar prądów fazowych silnika z magnesami trwałymi", *Maszyny Elektryczne - Zeszyty Problemowe*, nr 4, str. 57-61, 2012.
- [2]. Microsemi: "Park, inverse Park and Clark, inverse Clark transformations MSS software implementation", *User Guide*, str. 5-10, 2013
- [3]. T. Rudnicki, R. Czerwiński, A. Fręchowicz: "Control drivers for PMSM motor", *Maszyny Elektryczne - Zeszyty Problemowe*, nr 90, str. 51-55, 2011.
- [4]. A.S.N. Mokhtar, M.B.I. Raez, M. Marufuzzman, M.A.M. Ali: "Hardware Implementation of a High Speed Inverse Park Transformation Using CORDIC and PLL for FOC Brushless Servo Drive", *Elektronika ir Elektrotehnika*, ISSN 1392-1215, vol. 19 nr 3, 2013.
- [5]. L. Jarzbowicz: "Error Analysis of Calculating Average d-q Current Components using Regular Sampling and Park Transformation in FOC Drives", *International Conference and Exposition on Electrical and Power Engineering (EPE)*, str. 901-905, 2014.
- [6]. J. Przepiórkowski: "Trójfazowy falownik wektorowy – zestaw ewaluacyjny TMS320F28035, część 1 Podstawy teoretyczne", *Elektronika Praktyczna*, nr 4, str. 109-112, 2010.
- [7]. R. Andraka: "A survey of CORDIC algorithms for FPGA based computers", Andraka Consulting Group, 16 Arcadia Drive North Kingstown, 1998
- [8]. T. Vladimirova and H. Tiggeler: "FPGA Implementation of Sine and Cosine Generators Using the CORDIC Algorithm", Surrey Space Centre University of Surrey.
- [9]. Z. Navabi: "Digital Design and Implementation with Field Programmable Devices" *Springer Science*, ISBN:1-4020-8011-5, str. 229-245, 2005.
- [10]. S. Mirzaei, A. Hosangadi, R. Kastner: "FPGA Implementation of High Speed FIR Filters Using Add and Shift Method", *In Proc. ICCD*, 2006.
- [11]. M. Petko, K. Gac, G. Karpel, G. Góra: "Acceleration of parallel robot kinematic calculations in FPGA", *IEEE International Conference on Industrial Technology*, e-ISBN: 978-1-4673-4568, str. 34-39, 2013.
- [12]. Altera: <http://www.altera.com>, 2017-01-15.

## Autorzy

mgr inż. Grzegorz Góra; Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie al. Mickiewicza 30, 30-059 Kraków [ggora@agh.edu.pl](mailto:ggora@agh.edu.pl);

dr inż. Grzegorz Karpel; Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie al. Mickiewicza 30, 30-059 Kraków [gkarpel@agh.edu.pl](mailto:gkarpel@agh.edu.pl);

mgr inż. Piotr Mars; Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie al. Mickiewicza 30, 30-059 Kraków [mars@agh.edu.pl](mailto:mars@agh.edu.pl);

Radosław Sitek; Energylandia, al. 3 Maja 2 32-640 Zator; [radoslawsitek@wp.pl](mailto:radoslawsitek@wp.pl)

Marek Goczał; Energylandia, al. 3 Maja 2 32-640 Zator; [marek@energy2000.pl](mailto:marek@energy2000.pl)