

Antoni KRAHEL

PROCEDURY ANALIZY HARMONICZNYCH DLA PROCESORÓW ARM

STRESZCZENIE *W artykule przedstawiono opis procedur Szybkiej Dyskretnej Transformacji Fouriera FFT, przygotowanych na procesory z rdzeniem ARM-9. Procedury te, używające arytmetyki stałoprzecinkowej, zostały napisane w asemblerze w sposób umożliwiający wykorzystanie ich w programach pisanych w języku C. Zostały one opracowane w celu wykorzystania w analizatorach mocy i jakości energii, zgodnych z obowiązującymi normami PN-EN 61000-4-7 i PN-EN 61000-4-30. W artykule zamieszczono również wyniki testów opracowanych procedur, pozwalających ocenić ich wydajność i dokładność. Uzyskane wyniki porównano z osiągnięciami analogicznych procedur napisanych bezpośrednio w języku C z wykorzystaniem standardowej biblioteki matematycznej, jak również z praktycznymi potrzebami.*

Słowa kluczowe: *analiza częstotliwościowa, FFT, DSP, ARM-9*

1. WSTĘP

Zarówno w przypadku pomiarów jakości energii w sieci prądu zmiennego, jak i badania kompatybilności urządzeń do tej sieci przyłączanych, zachodzi potrzeba pomiaru harmonicznym i interharmonicznym napięć i prądów sieciowych. W obu przypadkach procedurę pomiarową definiuje norma PN-EN 61000-4-7 [5]. Zaleca ona stosowanie rozkładu mierzonych przebiegów w sinusowy szereg Fouriera o postaci:

$$f(t) = C_0 + \sum_{m=1}^{\infty} C_m \sin\left(\frac{m}{M} \omega_1 t + \varphi_m\right), \quad (1)$$

gdzie:

$f(t)$ – przebieg czasowy analizowanego napięcia lub prądu;
 C_0 – składowa stała;

dr inż. Antoni KRAHEL

e-mail: antoni.krahel@iel.gda.pl

Institut Elektrotechniki, Oddział w Gdańsku

PRACE INSTYTUTU ELEKTROTECHNIKI, zeszyt 262, 2013

C_m – amplituda m -tej składowej sinusoidalnej o pulsacji ω_m :

$$\omega_m = \frac{m}{M} \omega_1, \quad (2)$$

φ_m – faza początkowa m -tej składowej;

ω_1 – pulsacja składowej podstawowej;

M – ilość analizowanych okresów składowej podstawowej.

We współczesnych cyfrowych analizatorach współczynniki szeregu wyznacza się zwykle za pomocą dyskretnej transformacji Fouriera (DFT) [4, 6]. Porównując sumę szeregu (1) z odwrotną dyskretną transformacją Fouriera (IDFT), zdefiniowaną jako:

$$\underline{x}(n) = \sum_{k=0}^{N-1} \underline{X}(k) e^{j \frac{2\pi kn}{N}}, \quad (3)$$

gdzie:

$\underline{x}(n)$ – ciąg próbek analizowanego sygnału (w dziedzinie czasu);

$\underline{X}(k)$ – dyskretna transformata Fouriera;

N – liczba analizowanych próbek,

łatwo zauważyć, że przy założeniu synchronicznego próbkowania, tj.

$$NT_p = MT_1, \quad (4)$$

gdzie:

T_p – okres próbkowania;

T_1 – okres składowej podstawowej analizowanego przebiegu,

oraz ograniczonego do połowy częstotliwości próbkowania widma analizowanego przebiegu, czyli:

$$C_m = 0 \quad \text{dla } m \geq \frac{N}{2M}, \quad (5)$$

współczynniki szeregu Fouriera: C_0 , C_m , φ_m można obliczyć z dyskretnej transformaty Fouriera $\underline{X}(k)$:

$$\underline{X}(k) = \frac{1}{N} \sum_{n=0}^{N-1} \underline{x}(n) e^{-j \frac{2\pi kn}{N}} \quad (6)$$

według wzorów:

$$C_0 = X(0), \quad (7)$$

$$C_m = 2|\underline{X}(mM)|, \quad (8)$$

$$\varphi_m = \frac{\pi}{2} + \arg \underline{X}(mM). \quad (9)$$

Co więcej, uwzględniając fakt, że DFT jest liniowym przekształceniem zespolonego ciągu wejściowego (w dziedzinie czasu) w zespolony ciąg wyjściowy (w dziedzinie częstotliwości), a analizowane przebiegi prądów i napięć są rzeczywiste, możliwe jest jednoczesne wyznaczenie dyskretnych transformat Fouriera [a tym samym współczynników szeregu (1)] dwóch analizowanych przebiegów (np. prądu i napięcia w danej fazie). W tym celu tworzy się z dwóch danych sygnałów rzeczywistych $x_1(n)$ i $x_2(n)$ sygnał zespolony w postaci:

$$\underline{x}(n) = x_1(n) + jx_2(n) \quad (10)$$

Sygnał ten poddawany jest transformacji (6), w wyniku której otrzymuje się transformatę, która zgodnie z zasadą superpozycji wynosi:

$$\underline{X}(k) = \underline{X}_1(k) + j\underline{X}_2(k). \quad (11)$$

Aby z niej wyodrębnić poszukiwane transformaty $\underline{X}_1(k)$ i $\underline{X}_2(k)$, korzysta się z zależności:

$$\underline{X}_1(k) = \frac{1}{2}(\operatorname{Re}\{\underline{X}_S(k)\} + j \operatorname{Im}\{\underline{X}_R(k)\}) \quad (12)$$

$$\underline{X}_2(k) = \frac{1}{2}(\operatorname{Im}\{\underline{X}_S(k)\} - j \operatorname{Re}\{\underline{X}_R(k)\}) \quad (13)$$

w których ciągi: sumacyjny \underline{X}_S i różnicowy \underline{X}_R wyrażają zależności:

$$\underline{X}_S(k) = \underline{X}(k) + \underline{X}(N-k) \quad (14)$$

$$\underline{X}_R(k) = \underline{X}(k) - \underline{X}(N-k). \quad (15)$$

Jak z przedstawionych zależności wynika, DFT jest bardzo użytecznym narzędziem do pomiarów harmonicznych w sieci zasilającej, tym bardziej, że istnieją bardzo wydajne algorytmy jej wyznaczania (FFT) [4]. Tradycyjnie do przetwarzania i analizy sygnałów wykorzystuje się procesory sygnałowe, posiadające architekturę zoptymalizowaną do tego typu zadań. Producenci tych procesorów często dostarczają również biblioteki gotowych do wykorzystania procedur cyfrowego przetwarzania sygnałów, w tym procedur FFT [8]. Procesory sygnałowe są jednak zwykle ubogo wyposażone we współczesne interfejsy komunikacyjne (USB, Ethernet, SD, MMC itp.), co utrudnia ich wykorzystanie w nowoczesnych przyrządach pomiarowych, od których wymaga się kompatybilności z innymi otaczającymi je urządzeniami, takimi jak: komputery, elektroniczne karty pamięci, urządzenia sieciowe itp. Alternatywą dla procesorów sygnałowych w tych zastosowaniach mogą być dynamicznie rozwijane w ostatnich latach procesory aplikacyjne ze standardowymi 32-bitowymi rdzeniami ARM [1-3]. Układy te są bogato wyposażone w interfejsy komunikacji z użytkownikiem i innymi urządzeniami i powszechnie wykorzystywane w drobnym przenośnym sprzęcie (komórki, aparaty cyfrowe, tablety itp.). Ustępują one pod względem mocy obliczeniowej procesorom sygnałowym, jednak w wielu zastosowaniach oferowana przez nie szybkość obliczeń może być wystarczająca. Producenci tych procesorów zwykle nie oferują dla nich bibliotek DSP zawierających procedury FFT. Wyko-

rzystanie ich do budowy przyrządów pomiarowych wielkości elektrycznych, w tym analizatorów mocy czy jakości energii, wiąże się z koniecznością opracowania takich bibliotek.

2. BIBLIOTEKA QFFT

Opracowana w Oddziale Gdańskim Instytutu Elektrotechniki biblioteka QFFT stanowi zbiór efektywnych, stałoprzecinkowych procedur Dyskretnej Transformacji Fouriera. Dla zapewnienia maksymalnej precyzji i wydajności, biblioteka w całości została stworzona w assemblerze procesorów ARM. Wykorzystuje ona listę rozkazów w wersji ARM v5E [2], a więc może być używana na procesorach wyposażonych w rdzenie ARM9 lub nowsze (ARM11, CortexA8 itd.). Zawarte w niej procedury są dostępne z poziomu programu pisanego w języku C [3] za pośrednictwem dwóch funkcji: `qfft` – obliczającej Dyskretną Transformację Fouriera przebiegu zespolonego i `qrfft2` – obliczającej jednocześnie transformaty dwóch ciągów rzeczywistych według zależności przedstawionych we wprowadzeniu (10-15). W języku „C” deklaracje tych funkcji posiadają postać:

```
void qfft(cqn *i, cqn *o, int w);
void qrfft2(qn *i1, qn *i2, cqn *o1, cqn *o2, int w);
```

gdzie:

- `i, i1, i2` – wskaźniki do wektorów zawierających analizowane przebiegi;
- `o, o1, o2` – wskaźniki do wektorów, w których mają być umieszczone wyniki analizy;
- `qn` – typ reprezentujący liczbę rzeczywistą w 32-bitowym formacie stałoprzecinkowym;
- `cqn` – typ reprezentujący liczbę zespoloną zdefiniowaną jako:


```
struct{ qn re, qn im },
```
- `w` – parametr określający rozmiar analizowanych przebiegów: N zgodnie z zależnością:

$$N = 2^W, \quad W = 3 \dots 12. \quad (16)$$

Przyjęte rozwiązanie pozwala na bardzo proste użycie zawartych w bibliotece procedur w programie napisanym w języku C, jak to przedstawiono na przykładzie fragmentu programu analizy napięć i prądów sieci trójfazowej:

```
#include<qfft.h>

qn ut[3][2048]; //próbki napięcia w 3 fazach
qn it[3][2048]; //próbki prądu w 3 fazach
```

```

cqn Uh[3][1024]; //widmo napięcia w 3 fazach
cqn Ih[3][1024]; //widmo prądu w 3 fazach
int i;

for(i=0; i<3; i++)
    qrfft2(ut[i], it[i], Uh[i], Ih[i], 11);

```

Dzięki starannemu projektowi, zrealizowanemu bezpośrednio w asemblerze, procedury zawarte w bibliotece charakteryzuje duża wydajność i dokładność obliczeń. Świadczą o tym wyniki przeprowadzonych eksperymentów.

W tabelach 1 i 2 przedstawiono zmierzone oscyloskopem czasy wykonania procedur: `qrfft` i `qrfft2` w zależności od rozmiaru analizowanych ciągów.

TABELA 1

Zestawienie czasów wykonania funkcji: `qrfft`, `ffft` i `dfft` dla różnych ilości punktów

Liczba punktów N	Czas wykonania funkcji [ms]:		
	qrfftN	ffftN	dfftN
8	0,010	0,077	0,152
16	0,025	0,200	0,397
32	0,058	0,506	0,983
64	0,130	1,20	2,35
128	0,292	2,82	5,45
256	0,652	6,48	12,5
512	1,46	14,6	27,9
1024	3,25	32,7	61,9
2048	7,10	71,9	136
4096	15,5	157,0	297

TABELA 2

Zestawienie czasów wykonania funkcji: `qrfft2`, `frfft2` i `drfft2` dla różnych ilości punktów

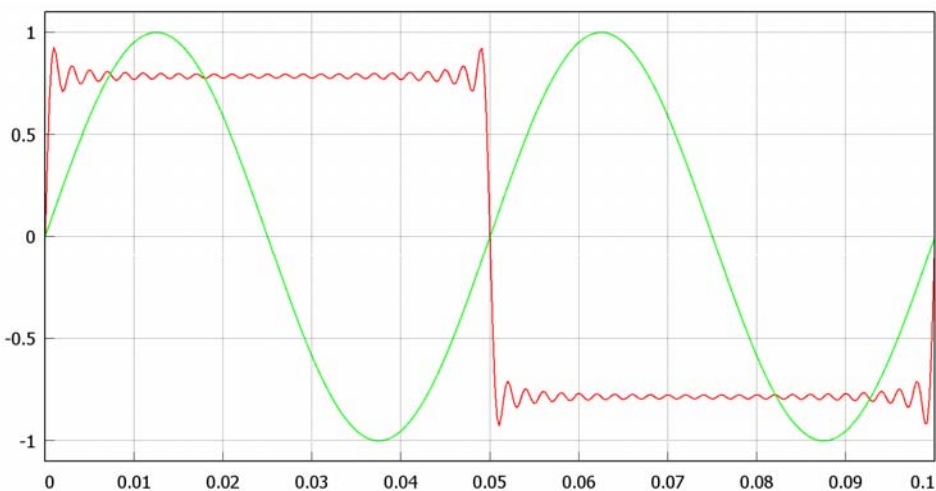
Liczba punktów N	Czas wykonania funkcji [ms]:		
	qrfft2N	frfft2N	drfft2N
8	0,014	0,085	0,169
16	0,031	0,216	0,440
32	0,067	0,530	1,10
64	0,148	1,29	2,61
128	0,324	3,02	6,06
256	0,738	6,90	13,6
512	1,65	15,4	30,5
1024	3,60	34,1	66,7
2048	7,86	74,7	146,3
4096	17,0	162,3	317,9

Dla porównania w sąsiednich kolumnach zamieszczono czasy wykonania analogicznych procedur napisanych bezpośrednio w języku C z wykorzystaniem standardowej biblioteki matematycznej. Funkcje `ffft()` i `frfft2()` zrealizowane zostały w pojedynczą precyzją obliczeń (`float`), a funkcje `dfft()` i `drfft2()` podwójną (`double`).

Wszystkie próby przeprowadzone zostały na tym samym module MMnet-1002 [7] produkcji Propox, zawierającym procesor z rdzeniem ARM-9 typu AT91SAM9260 [1] (Atmel), taktowany zegarem 200 MHz, oraz zewnętrzną pamięć SDRAM, taktowaną zegarem 100 MHz. W trakcie prób program, dane i stos były ulokowane w zewnętrznej pamięci SDRAM na wspólnej stronie.

Jak łatwo zauważyć w tabelach: 1 i 2, przy większej ilości punktów ($N \geq 512$) czasy wykonania opracowanych procedur: `qfft` i `qrfft2` są około dziesięciokrotnie krótsze od ich zmiennoprzecinkowych odpowiedników, wykorzystujących pojedynczą precyzję, oraz około dwudziestokrotnie krótsze od procedur, wykorzystujących podwójną precyzję obliczeń. Dla mniejszej liczby punktów ta różnica maleje do około siedmiu i czternastu razy w przypadku procedur ośmiopunktowych.

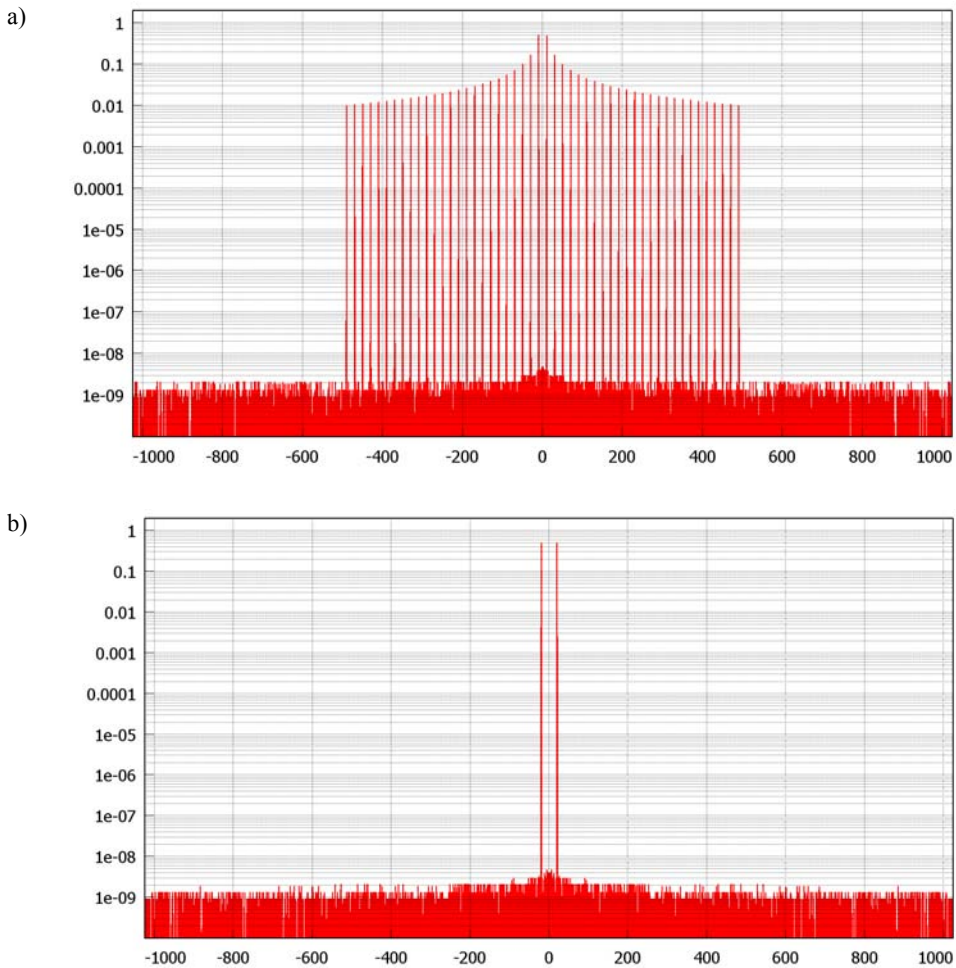
Sprawdzenia dokładności procedur zawartych w bibliotece QFFT dokonano w oparciu o wyniki wykonanych za ich pomocą transformacji specjalnie przygotowanych sygnałów wzorcowych: rzeczywistych i zespolonych. Dla wszystkich testowanych procedur uzyskano podobne wyniki, dlatego też w niniejszej publikacji ograniczono się do przedstawienia wyników jednej próby 2048-punktowej procedury `qrfft2`. Polegała ona na transformacji dziesięciookresowego odcinka dwóch sygnałów rzeczywistych, oznaczonych jako: s_1 i s_2 . Sygnał s_1 był sygnałem prostokątnym zszyntezowanym z pierwszych pięćdziesięciu harmonicznych, zaś s_2 sygnałem sinusoidalnym o częstotliwości drugiej harmonicznej (rys. 6).



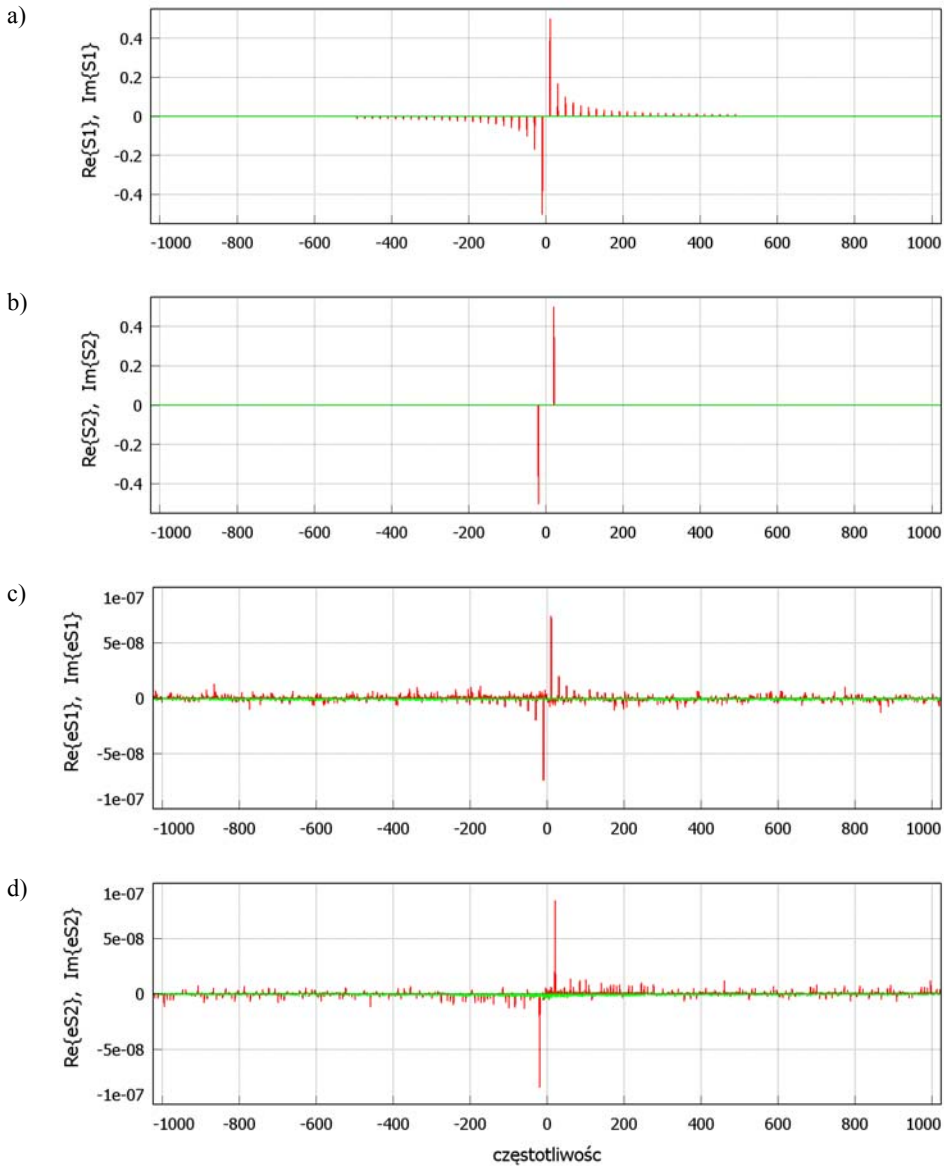
Rys. 6. Sygnały testowe: s_1 – prostokątny zszyntezowany z 50-ciu harmonicznych (czerwony), s_2 – sinusoidalny o częstotliwości drugiej harmonicznej (zielony). Czas unormowany jest do okresu analizy (1 odpowiada 10 okresom s_1)

Użycie do próby zsyntezowanego sygnału prostokątnego zamiast idealnego pozwoliło uniknąć błędów spowodowanych efektem utożsamiania składowych o częstotliwościach wyższych od połowy częstotliwości próbkowania (sygnał o ostro ograniczonym paśmie częstotliwości). Z kolei wybór drugiego sygnału jako sinusoidalnego o podwojonej częstotliwości, czyli takiej, która nie występuje w widmie sygnału prostokątnego, wynikał z potrzeby zaobserwowania ewentualnych przesłuchów pomiędzy przetwarzanymi sygnałami.

Na rysunku 7 przedstawiono w skali logarytmicznej wykresy modułów uzyskanych w wyniku wykonania procedury `qrfft2` transformat, zaś na rysunku 8 wykresy ich części rzeczywistych i urojonych wraz z błędami ich wyznaczenia.



Rys. 7. Moduły transformat Fouriera sygnałów testowych uzyskanych za pomocą 2048-punktowej procedury `qrfft2`: a) s_1 , b) s_2 . Częstotliwość unormowana jest do odwrotności czasu analizy (1 odpowiada 0,1 częstotliwości s_1)



Rys. 8. Rezultaty transformacji za pomocą funkcji qrfft2: a) transformata sygnału s_1 , b) transformata sygnału s_2 , c) błąd wyznaczenia transformaty sygnału s_1 , d) błąd wyznaczenia transformaty sygnału s_2 . Częstotliwość unormowana jest do odwrotności czasu analizy (10 odpowiada częstotliwości s_1)

Dla zapewnienia naturalnego (ciągłego) wyglądu wykresów wyznaczonych widm analizowanych sygnałów, oś częstotliwości dyskretnych transformat Fouriera rozciąto w punkcie przeciwnym do zera, uzyskując po rozwinięciu zakres unormowanej częstotliwości, wynoszący $-1024 \div 1023$.

Na wszystkich wykresach wyraźnie widoczne są prążki, odpowiadające wyłącznie harmonicznemu występującym w transformowanych sygnałach. Na żadnym z nich nie widać śladu prążków drugiego sygnału (przesłuch) ani prążków o innych częstotliwościach (intermodulacja). Na wykresach modułów wykreślonych w skali logarytmicznej widoczny jest poziom szumów poniżej $1 \cdot 10^{-8}$. Z kolei na wykresach błędów części rzeczywistej i urojonej transformacji widać, że nie przekraczają one poziomu $1 \cdot 10^{-7}$.

W celu porównania uzyskanych wyników z praktycznie występującymi potrzebami posłużono się wymaganiami, jakie stawia konstrukcja trójfazowego analizatora harmonicznych klasy I. Przyrząd taki zgodnie z normą PN-EN 61000-4-7 powinien wykonywać pięć razy na sekundę analizy dziesięciookresowych wycinków mierzonych przebiegów w zakresie częstotliwości do 50-tej harmonicznej. W stanie ustalonym, przy przebiegach napięcia o pojedynczej częstotliwości, powinien on zapewnić dokładność pomiarów lepszą niż 5% wartości mierzonej dla sygnałów większych niż 1% wartości nominalnej i 0,05% wartości nominalnej dla sygnałów mniejszych. Wymagania dla pomiarów prądu pominięto, gdyż są nieco łagodniejsze. Określony w normie dopuszczalny błąd dotyczy całego toru pomiarowego przyrządu. Błąd wprowadzany przez samą procedurę analizy harmonicznych powinien stanowić jedynie niewielką jego część (np. 10%), ponieważ należy założyć, że jego głównym źródłem będą wejściowe układy analogowe. Ze względów praktycznych należy założyć, że transformacje DFT nie powinny absorbować więcej niż 25% czasu pracy procesora zastosowanego w przyrządzie (przyrząd jednoprocessorowy).

Jak łatwo zauważyć, zaprezentowane funkcje charakteryzują się dokładnością wielokrotnie wyższą od wymaganej. Znacznie bardziej krytyczne są wymagania dotyczące czasów analizy. Przyjmując optymalną częstotliwość próbkowania wynoszącą ok. 10,24 kHz, w każdym cyklu pomiarowym (co 200 ms) należy poddawać analizie sześć ciągów po 2048 rzeczywistych próbek (10 okresów sieci). Użycie do tego celu 2048-punktowej procedury `qrfft2` (trzy wywołania na sześć przebiegów) absorbuje ok. 23,6 ms na 200 ms, czyli 11,8% czasu pracy procesora. Dla porównania, realizacja tego samego zadania za pomocą zmiennoprzecinkowej wersji procedury o pojedynczej precyzji wymagałoby już ponad 112% mocy procesora, co oczywiście z góry wyklucza możliwość takiego rozwiązania.

3. PODSUMOWANIE

Dyskretna Transformacja Fouriera jest użytecznym narzędziem analizy sygnałów, coraz częściej wykorzystywanym w elektrotechnice. Tradycyjnie do DFT wykorzystuje się procesory sygnałowe. Dynamiczny rozwój w ostatnich latach procesorów aplikacyjnych może to częściowo zmienić. Procesory te oferują znacznie bogatszy i bardziej dostosowany do współczesnych potrzeb zestaw interfejsów komunikacyjnych i wymagają przy tym znacznie mniej energii do zasilania, co ułatwia ich wykorzystanie w urządzeniach przenośnych (baterijnych). Jednak jak pokazano w niniejszym artykule, można stosunkowo łatwo dla tych procesorów opracować biblioteki procedur DFT, które będą dostatecznie dokładne i wydajne dla typowych pomiarów przebiegów prądów i napięć w sieciach energetycznych.

LITERATURA

1. Atmel: AT91 ARM Thumb Microcontrollers AT91SAM9260 (DOC6221). Atmel, 2009.
2. ARM: ARM and Thumb-2 Instruction Set Quick Reference Card (QRC0001M). ARM, 2008.
3. ARM: Procedure Call Standard for the ARM Architecture (ARM IHI 0042D) ARM, 2009.
4. Oppenheim A. V., Szafer R. W.: Cyfrowe przetwarzanie sygnałów. WKŁ, Warszawa, 1979.
5. PN-EN 61000-4-7 Kompatybilność elektromagnetyczna (EMC) Część 4-7: Metody badań i pomiarów. Ogólny przewodnik dotyczący pomiarów harmonicznych i interharmonicznych oraz przyrządów pomiarowych dla sieci zasilających i przyłączonych do nich urządzeń. PKN, Warszawa, 2007.
6. PN-EN 61000-4-30 Kompatybilność elektromagnetyczna (EMC) Część 4-30: Metody badań i pomiarów Metody pomiaru jakości energii. PKN, Warszawa 2005.
7. Propox: MMnet1001 Moduł z mikrokontrolerem ARM9 i Ethernetem. Instrukcja użytkownika. Gdańsk, 2008.
8. Texas Instruments: FFT Library Module user's Guide C28x Foundation Software. Texas Instruments Inc., 2002

Rękopis dostarczono dnia 22.05.2013 r.

FFT PROCEDURES PACKAGE FOR PROCESSORS WITH ARM CORE

Antoni KRAHEL

ABSTRACT *In this article the description of FFT procedures prepared for processors with ARM-9 core is presented. These procedures using the fixed point arithmetic were written in Assembly language to programs written in C language developed. They were prepared for use in the power and quality of energy analysers according to standards: PN-EN 61000-4-7 and PN-EN 61000-4-30. The article also provides the results testing the developed procedures to evaluate their performance and accuracy. The results obtained were compared with similar procedures written directly in C, as well as with practical needs.*

Keywords: *FFT, DSP, ARM-9*



Dr inż. Antoni KRAHEL – absolwent Wydziału Elektroniki Politechniki Gdańskiej, od 1993 r. pracuje w Oddziale Gdańskim Instytutu Elektrotechniki, gdzie w 2006 r. uzyskał stopień doktora nauk technicznych. Obecnie zatrudniony na stanowisku adiunkta. Zajmuje się wykorzystaniem technik cyfrowego przetwarzania sygnałów w metrologii elektrycznej i energoelektronice.