

Galina ȚARIOVA, Alexandr ȚARIOV
POLITECHNIKA SZCZECIŃSKA, WYDZIAŁ INFORMATYKI

Struktury algorytmiczne jednostek procesorowych do realizacji bazowych operacji dyskretnej transformaty falkowej

Mgr Galina ȚARIOVA

Ukończyła studia na Wydziale Matematyki i Cybernetyki Mołdawskiego Uniwersytetu Państwowego w Kiszyniowie w 1978 r. Jej zainteresowania naukowe są związane z różnymi aspektami aplikacyjnymi matematyki oraz informatyki teoretycznej i stosowanej: analizą falkową, metodami numerycznymi, matematyką dyskretną, algorytmami cyfrowego przetwarzania sygnałów.



e-mail: gtariova@wi.ps.pl

Dr hab. inż. Alexandr ȚARIOV

Ukończył studia na Wydziale Automatyki i Urządzeń Obliczeniowych Uniwersytetu Miernictwa (Sewastopol), obronił pracę doktorską w 1984 r., habilitacyjną - w 2001 r. Jest profesorem w Instytucie Architektury Komputerów i Telekomunikacji na Wydziale Informatyki Politechniki Szczecińskiej. Jego zainteresowania naukowe to algorytmy cyfrowego przetwarzania oraz transmisji sygnałów, sprzętowe wspomaganie oraz zrównoleglenie obliczeń.



e-mail: atariov@wi.ps.pl

Streszczenie

W pracy zostało przedstawione podejście do zoptymalizowanej organizacji struktur algorytmicznych jednostek obliczeniowych dla realizacji bazowych operacji FDWT/IDWT ze zredukowaną liczbą mnożeń (lub układów mnożących w przypadku implementacji sprzętowej). Podejście to pozwala zmniejszyć nakłady obliczeniowe, zapotrzebowanie na zasoby sprzętowe oraz stworzyć dogodny warunki do efektywnej realizacji metod falkowego przetwarzania danych w układzie reprogramowalnym.

Słowa kluczowe: dyskretna transformata falkowa DWT, szybkie algorytmy, procesory DWT.

Algorithmic structures of processing units for IDWT basic operations implementation

Abstract

This paper is concerned with the novel algorithmic structures for the realization of FDWT and IDWT basic procedures with the reduced number of arithmetic operations. As to well-known approaches, the immediate implementation of the above procedures requires $2L$ multipliers both for the DWT and IDWT basic procedures plus $2(L-1)$ adders for DWT and L adders for IDWT. At the same time, proposed algorithms require only $1\frac{1}{2} \cdot L$ multipliers for the both procedures plus $2L-1$ adders for FDWT and $L+1$ adders for IDWT basic procedures. The proposed structures can be successfully applied to accelerate calculations in the FPGA-based platforms as well as to enhance the efficiency of hardware in general.

Keywords: discrete wavelet transform, fast algorithms, DWT processors.

1. Wstęp

Dyskretna transformata falkowa (Discrete Wavelet Transform - DWT) jest obecnie podstawowym narzędziem cyfrowego przetwarzania sygnałów [1-4].

Procedura obliczeniowa prowadząca do efektywnej realizacji DWT nosi nazwę piramidy Mallata. Proces tworzenia wielopoziomowej reprezentacji według tego algorytmu jest iteracyjny. W każdej iteracji sygnał jest przedstawiany w postaci sumy reprezentacji szczegółowej i zgrubnej, a na każdym kolejnym poziomie reprezentacja zgrubna z poziomu poprzedniego zostaje znów przedstawiona jako suma reprezentacji szczegółowej i zgrubnej [1, 2].

„Cegiełką” całej tej procedury jest „bazowa operacja DWT” - operacja mnożenia wektora przez macierz bazy DWT, opisującej współczynniki filtrów [3, 4].

W postaci wektorowo-macierzowej operację bazową FDWT (Forward Discrete Wavelet Transform) można zdefiniować w następujący sposób:

$$\mathbf{Y}_{2 \times 1}^{(l)} = \mathbf{F}_{2 \times L} \cdot \mathbf{X}_{L \times 1}^{(l)}, \quad l = 0, \frac{N}{2^{k+1}} - 1. \quad (1)$$

gdzie N - liczba próbek sygnału pierwotnego, L - rozmiar ruchomego okna, definiującego część sygnału przetwarzaną za pomocą danej operacji bazowej, $k = 0, K-1$ - numer kroku dekompozycji określający poziom detalizacji analizy falkowej, K - całkowita liczba kroków dekompozycji.

Macierz $\mathbf{F}_{2 \times L} = \begin{bmatrix} h_0 & h_1 & \dots & h_{L-1} \\ g_0 & g_1 & \dots & g_{L-1} \end{bmatrix}$ we wzorze (1) jest

macierzą bazy DWT o wymiarach $(2 \times L)$, której elementy reprezentują współczynniki filtrów, odpowiednio dolnoprzepustowego i górnoprzepustowego. Będziemy nazywać tę macierz „macierzą bazy DWT”, natomiast operację mnożenia tej macierzy przez wektor odpowiednich danych nazwiemy „bazową operacją DWT”.

Wektor $\mathbf{X}_{L \times 1}^{(l)} = [x_{2l}, x_{2l+1}, \dots, x_{L+2l-1}]^T$ we wzorze (1) przedstawia wektor-kolumnę rzędu L , którego elementy reprezentują odpowiedni zestaw próbek sygnału.

$\mathbf{Y}_{2 \times 1}^{(l)} = [y_{2l}, y_{2l+1}]^T$ - dwuelementowy wektor danych wyjściowych dla każdej operacji bazowej (1), w którym y_{2l} - jest odpowiednim współczynnikiem DWT, a y_{2l+1} - wynikiem aproksymacji, podlegającym razem z odpowiednimi wynikami innych operacji (o nieparzystej wartości indeksu dolnego) dalszej transformacji.

Bazowa operacja IDWT (Inverse Discrete Wavelet Transform), czyli odwrotna DWT, przybiera wówczas następującą postać:

$$\mathbf{X}_{L \times 1}^{(l)} = \mathbf{F}_{2 \times L}^T \cdot \mathbf{Y}_{2 \times 1}^{(l)}. \quad (2)$$

Obliczenie (1) wymaga wykonania $2L$ mnożeń i $2(L-1)$ dodawań, zaś obliczenie (2) odpowiednio - $2L$ mnożeń oraz L dodawań [5, 6].

Aż do czasów obecnych nie istniało praktyczne, a jednocześnie uniwersalne rozwiązanie pozwalające zredukować liczbę operacji arytmetycznych przy realizacji operacji bazowych FDWT (IDWT) dla dowolnego zestawu oraz dowolnych długości filtrów falkowych. Prawdopodobnie dopiero w pracach [7-10] kwestia ta znalazła jakieś-to praktyczne rozwiązanie.

W poprzednim artykule autorów opisana została propozycja redukcji liczby mnożeń (lub bloków mnożących) przy realizacji bazowych operacji FDWT (IDWT) kosztem zwiększenia liczby dodawań (symatorów) [9]. Okazuje się, że istnieje możliwość zastosowania bardziej racjonalnego podejścia, pozwalającego zmniejszyć liczbę mnożeń w badanych algorytmach nie zwiększając liczby dodawań. Opracowaniu takich algorytmów i struktur jednostek procesorowych oraz analizie ich efektywności poświęcony jest ten artykuł.

2. Synteza struktur jednostek procesorowych do realizacji bazowych operacji FDWT/IDWT ze zredukowaną liczbą bloków mnożących

Na podstawie elementów macierzy $\mathbf{F}_{2 \times L}$ zbudujemy nową macierz diagonalną:

$$\mathbf{D}_{\frac{3L}{2}} = \bigoplus_{i=0}^{\frac{L-1}{2}} \mathbf{D}_3^{(i)},$$

gdzie podmacierz

$$\mathbf{D}_3^{(i)} = \text{diag}[(h_{(L-1)-2i} - h_{2i}), (h_{(L-1)-2i} + h_{2i}), h_{2i}]$$

też jest macierzą diagonalną o rozmiarze (3×3) .

Wprowadźmy teraz trzy macierze sumowania:

$\mathbf{A}_{\frac{3L}{2} \times L} = (\mathbf{I}_{\frac{L}{2}} \otimes \mathbf{T}_{3 \times 2})$ - macierz dodawań poprzedzających

mnożenie; $\mathbf{A}_{L \times \frac{3L}{2}} = (\mathbf{I}_{\frac{L}{2}} \otimes \mathbf{T}_{2 \times 3})$ - macierz dodawań następujących po mnożeniu;

$\mathbf{A}_{3 \times \frac{3L}{2}} = (\mathbf{I}_{1 \times \frac{L}{2}} \otimes \mathbf{I}_3)$ - macierz sumowania

wyników cząstkowych, gdzie \mathbf{I} - macierz jednostkowa o wymiarze określonym za pomocą dolnego indeksu,

$$\mathbf{T}_{3 \times 2} = \begin{bmatrix} 1 & & \\ & 1 & \\ 1 & & -1 \end{bmatrix}, \quad \mathbf{T}_{2 \times 3} = \begin{bmatrix} & 1 & 1 \\ 1 & & \end{bmatrix};$$

\otimes - symbol iloczynu Kroneckera, $\bigoplus_{i=1}^n$ - symbol sumy prostej n

elementowej [11-12], $\mathbf{I}_{k \times m}$ - macierz składająca się z jedynek o wymiarze określonym za pomocą dolnego indeksu [12].

Zbudujemy również monomialną macierz tasowania danych $\mathbf{P}_L = \|p_{ij}\|$, której niezerowe elementy są zdefiniowane w sposób następujący:

$$\begin{cases} p_{2i+1, 2i+1} = 1, i = 0, \frac{L}{2} - 1, \\ p_{2(i+1), L-2i} = 1, i = 0, \frac{L}{2} - 1 \end{cases}$$

Uwzględniając wprowadzone konstrukcje wektorowo-macierzowe, algorytmiczną przestrzenno-czasową strukturę procesu realizacji bazowej operacji FDWT można przedstawić następująco:

$$\mathbf{Y}_{2 \times 1}^{(l)} = \mathbf{T}_{2 \times 3} \mathbf{A}_{3 \times \frac{3L}{2}} \mathbf{D}_{\frac{3L}{2}} \mathbf{A}_{\frac{3L}{2} \times L} \mathbf{P}_L \mathbf{X}_{L \times 1}^{(l)}. \quad (3)$$

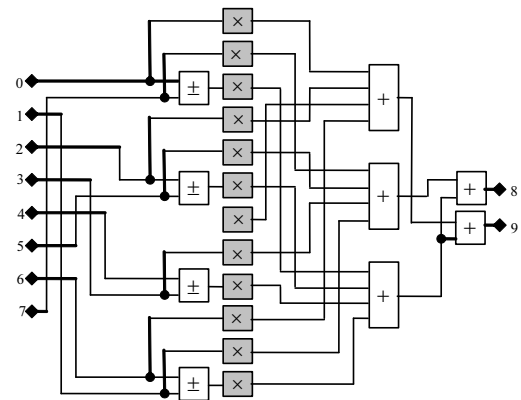
Na rysunku 1 została pokazana struktura jednostki procesorowej, realizującej operację bazową FDWT dla filtrów Daub-8 zgodnie z opracowaną procedurą. Implementacja sprzętowa tej struktury będzie wymagała czterech 2-wejściowych sumatorów-substraktorów, dwóch sumatorów 2-wejściowych, trzech sumatorów 3-wejściowych a także dwunastu bloków mnożących, lecz w tym przypadku - przez liczby stałe, czyli nietypowych, jedno-wejściowych. Bloki te mogą być zrealizowane m.in. w postaci zwykłych szyfratorów (co też jest poważnym atutem) i żeby odróżnić je od mnożarek konwencjonalnych zostały one na rysunku lekko zabarwione.

Przypomnijmy, iż wspomniane liczby stałe dla filtrów Daub-8 zdefiniowane jako elementy macierzy bazy DWT, która w przypadku filtrów Daub-8 przybiera następującą postać:

$$\mathbf{F}_{2 \times 8} = \begin{bmatrix} c_0 & c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & c_7 \\ c_7 & -c_6 & c_5 & -c_4 & c_3 & -c_2 & c_1 & -c_0 \end{bmatrix},$$

przy czym:

$$\begin{aligned} c_0 &= 0.2303778133, & c_1 &= 0.7148465706, & c_2 &= 0.6308807679, \\ c_3 &= -0.0279837694, & c_4 &= -0.1870348117, & c_5 &= 0.0308413818, \\ c_6 &= 0.0328830117, & c_7 &= -0.0105974018. \end{aligned}$$



Rys. 1. Struktura jednostki procesorowej realizującej bazową operację FDWT według procedury (3) dla przykładu filtrów Daub-8

Fig. 1. The processor unit structure for the implementation of the FDWT basic operation corresponding to (3) for Daub-8 filters type

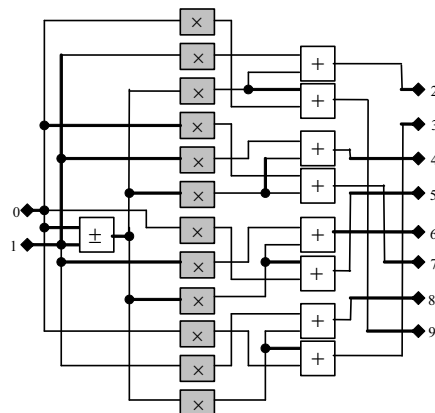
Dla bazowej operacji IDWT procedura obliczeniowa przybiera następującą postać:

$$\mathbf{X}_{L \times 1}^{(l)} = \mathbf{P}_L \mathbf{A}_{L \times \frac{3L}{2}} \mathbf{D}_{\frac{3L}{2}} \mathbf{P}_{\frac{3L}{2} \times 3} \mathbf{T}_{3 \times 2} \mathbf{Y}_{2 \times 1}^{(l)}, \quad (4)$$

gdzie

$$\mathbf{P}_{\frac{3L}{2} \times 3} = \mathbf{A}_{3 \times \frac{3L}{2}}^T.$$

Na rysunku 2 została przedstawiona struktura jednostki procesorowej, realizującej operację bazową IDWT dla filtrów Daub-8 według proponowanego podejścia.



Rys. 2. Struktura jednostki procesorowej realizującej bazową operację IDWT według procedury (4) dla przykładu filtrów Daub-8

Fig. 2. The processor unit structure for the implementation of the IDWT basic operation corresponding to (4) for Daub-8 filters type

Jednostka do realizacji IDWT będzie składać się z ośmiu 2-wejściowych sumatorów, jednego 2-wejściowego sumatora-substraktora oraz tak samo jak w poprzedniej strukturze dwunastu analogicznych układów mnożenia przez liczbę stałą.

Porównajmy teraz proponowane w niniejszym artykule struktury ze strukturami wynikającymi z podejścia opisanego w artykule [9]. Należy zwrócić uwagę, iż, mimo że wszystkie syntezowane struktury posiadają zredukowaną liczbą bloków mnożących względem bezpośredniej (tzw. naiwnej) równoległej implementacji sprzętowej, różnice w sposobach ich organizacji powodują indywidualną przydatność implementacyjną owych struktur. Przedstawione w tej pracy struktury mogą (ale nie muszą) zostać z powodzeniem zaimplementowane w takich układach reprogramowalnych, które nie zawierają tradycyjnych bloków mnożących, ponieważ wymagają bloków mnożenia przez wartość stałą, czyli kodoprzetworników (szyfratorów). Takie bloki mogą być dość efektywnie zrealizowane w FPGA z tradycyjną bramkową strukturą.

Natomiast, jeśli w układzie reprogramowalnym są już zainstalowane przez producenta konwencjonalne „mnożarki”, prawdopodobnie warto użyć rozwiązania strukturalne z pracy [9] bazujące na zastosowaniu standardowych 2-wejściowych układów mnożących.

3. Ocena zasobów sprzętowych

W tabelach 1 oraz 2 zostały przedstawione liczby bloków mnożenia i dodawania dla klasycznych oraz proponowanych struktur do realizacji operacji bazowych FDWT (IDWT) dla różnych długości L odpowiedzi impulsowych filtrów.

Tab. 1. Oszacowanie liczby bloków operacyjnych w zależności od długości L odpowiedzi impulsowych filtrów dla proponowanej oraz klasycznej struktury do realizacji operacji bazowej **FDWT**

Tab. 1. Estimation of the number of processing units depending on the filter length L available for the proposed and classical structure realizing FDWT basic operation

L	„mnożarki”		„sumatory”	
	kl.	prop.	kl.	prop.
4	8	6	6	7
6	12	9	10	11
8	16	12	14	15
10	20	15	18	19
12	24	18	22	23
14	28	21	26	27
16	32	24	30	31
18	36	27	34	35
20	40	30	38	39
22	44	33	42	43
24	48	36	46	47

Tab. 2. Oszacowanie liczby bloków operacyjnych w zależności od długości L odpowiedzi impulsowych filtrów dla proponowanej oraz klasycznej struktury do realizacji operacji bazowej **IDWT**

Tab. 2. Estimation of the number of processing units depending on the filter length L available for the proposed and classical structure realizing IDWT basic operation

L	„mnożarki”		„sumatory”	
	kl.	prop.	kl.	prop.
4	8	6	4	5
6	12	9	6	7
8	16	12	8	9
10	20	15	10	11
12	24	18	12	13
14	28	21	14	15
16	32	24	16	17
18	36	27	18	19
20	40	30	20	21
22	44	33	22	23
24	48	36	24	25

Jak widać, zaproponowane w artykule struktury algorytmiczne pozwalają zredukować łączną liczbę bloków mnożących względem metody „klasycznej”. Liczba bloków mnożących w obu opracowanych strukturach wynosi $1,5 \cdot L$, natomiast liczba dwuwejściowych sumatorów - w przypadku realizacji operacji bazowej FDWT - $(2L-1)$, zaś w przypadku realizacji operacji bazowej IDWT - $(L+1)$. Jeśli chodzi o stopień redukcji dla poszczególnych typów bloków, to liczba bloków mnożących nie-

zbędnych dla realizacji opracowanych w artykule algorytmów jest w przybliżeniu o 25% mniejsza względem pierwowzoru, natomiast liczba dwuwejściowych sumatorów jest o jeden blok większa.

Wprowadźmy współczynnik k stosunku „kosztu” τ_x implementacji operacji mnożenia względem „kosztu” implementacji operacji dodawania τ_+ . Koszt w tym przypadku może przybierać znaczenie ilości zasobów sprzętowych (np. liczby bramek logicznych) niezbędnych dla implementacji bloku mnożącego (sumatora).

Niech $\tau_x = k \cdot \tau_+$. Wtedy zysk implementacji struktury algorytmicznej proponowanej jednostki FDWT względem bezpośredniej (naiwnej) implementacji wzoru (1) dla dowolnego L będzie określony następującym wzorem:

$$Z_{FDWT} = \frac{2(kL + L - 1)}{1,5kL + 2L - 1},$$

Natomiast zysk implementacji proponowanej struktury algorytmicznej jednostki IDWT względem bezpośredniej (naiwnej) implementacji wzoru (2) dla dowolnego L będzie określony następującym wzorem:

$$Z_{IDWT} = \frac{L(2k + 1)}{L(1,5k + 1) + 1}.$$

4. Podsumowanie

W artykule zaproponowano nowe rozwiązania do zoptymalizowanej sprzętowo implementacji operacji bazowych FDWT oraz IDWT. Oczywiście jest, że w podobny sposób mogą być skonstruowane efektywne struktury jednostek procesorowych do realizacji bazowych operacji dyskretnej transformaty falkowej dla innych typów filtrów oraz innych zestawów współczynników wagowych.

5. Literatura

- [1] J. T. Białasiewicz, Falki i aproksymacje, WNT, Warszawa, 2000.
- [2] T. Zieliński, Cyfrowe przetwarzanie sygnałów. Od teorii do zastosowań. WKŁ, Warszawa 2005.
- [3] S. G. Mallat, A theory for multiresolution signal decomposition: The wavelet representation, IEEE Trans. Patt. Anal. Mach. Intell., vol. 11, pp. 674-693, July 1989.
- [4] I. Daubechies, Ten Lectures on Wavelets, ser. no. 61 in CBMS-NSF Series in Applied Mathematics. Philadelphia, PA: SIAM, 1992.
- [5] C. Chakrabarti, M. Vishwanath, and R. M. Owens, Architectures for wavelet transforms: A survey, J. VLSI Signal Processing, vol. 14, pp. 171- 192, 1996.
- [6] M. Weeks, M. Bayoumi, Discrete Wavelet Transforms: Architectures, Design and Performance Issues, Journal of VLSI Signal Processing, 2003, no. 35, pp. 155-178.
- [7] A. Țariov, G. Țariova, Synteza szybkich algorytmów implementacji bazowej operacji dyskretnej transformaty falkowej, Materiały III Krajowej Konferencji Elektroniki „KKE-2004”, Kołobrzeg, czerwiec 2004.: Wydawnictwo Politechniki Koszalińskiej, tom I/2, Koszalin 2004, str. 297-302.
- [8] G. Țariova, A. Țariov, Nowe algorytmy realizacji bazowych operacji DWT ze zredukowaną liczbą mnożeń. Metody informatyki stosowanej. Roczniki informatyki stosowanej WIPS, Nr 9, Szczecin, Informa, 2005.
- [9] A. Țariov, G. Țariova, Zorientowane sprzętowo algorytmy realizacji bazowych operacji dyskretnej transformaty falkowej. - w czasopiśmie „Pomiary, Automatyka, Kontrola”, 2006, No 7a .
- [10] A. Țariov, T. Mąka, R. Maciaszczyk, G. Țariova. Struktury jednostek przetwarzających dla procesorów DWT. Materiały VIII Krajowej Konferencji Naukowej Reprogramowalne układy cyfrowe „RUC-2005”, 13-14 maja 2005, Szczecin, Informa, 2005.
- [11] E. Dagman., G. Kukharev. Szybkie dyskretne transformaty ortogonalne, Wydawnictwo Nauka, 1983.
- [12] A. Țariov, Modele algorytmiczne i struktury wysokowydajnych procesorów cyfrowej obróbki sygnałów, Szczecin, Informa, 2001.