

Andrzej DRYGAJŁO

MODUŁOWA POSTAĆ ALGORYTMÓW SZYBKIEJ TRANSFORMACJI WALSHA

Streszczenie. W pracy przeanalizowano algorytmy szybkiej transformacji Walsh'a pod kątem widzenia ich modułowości. Przedstawiono odpowiadające im struktury drzew pozwalające na bieżące równoległe wykonywanie podstawowych operacji motylkowych złożonych z operacji dodawania i odejmowania. Rozpatrzono dwie odmiany wielostopniowego przetwarzania, jedną wykorzystującą opóźnienia czasowe między próbkami sygnału, drugą opartą na zmianie częstotliwości próbkowania w kolejnych stopniach przetwarzania. Liczba operacji arytmetycznych wymagana dla otrzymania bieżącej N -punktowej transformaty jest proporcjonalna do $N \log_2 N$. Podano struktury dla naturalnego, diadycznego i sekwencyjnościowego uporządkowania w algorytmach szybkiej transformacji Walsh'a i porównano je z odpowiadającymi im strukturami algorytmów szybkiej transformacji Fouriera.

1. Wprowadzenie

Praktyczna przydatność dyskretnych transformacji, przede wszystkim dla potrzeb cyfrowego przetwarzania sygnałów, jest uwarunkowana możliwością budowy skutecznych algorytmów obliczeniowych do wyznaczania prostych i odwrotnych transformat. Z tego powodu istotne znaczenie mają algorytmy obliczeniowe redukujące liczbę operacji mnożenia i dodawania, które przyjęto nazywać algorytmami szybkich transformacji [1]. Do najbardziej efektywnych należą algorytmy szybkiej transformacji Walsh'a (FWT - Fast Walsh Transform) wymagające jedynie $N \log_2 N$ operacji dodawania i odejmowania [2]. Oprócz złożoności obliczeniowej coraz częściej rozpatrywana jest modułowość i możliwość bieżącego wykonywania algorytmów szybkich transformacji [3][4][5]. Struktury zapewniające modułowość algorytmów szybkich transformacji można złożyć ze struktur prostych filtrów cyfrowych. Podejście to, wykorzystane w przypadku algorytmów szybkiej transformacji Fouriera (FFT - Fast Fourier Transform), oparte jest na rekursywnej faktoryzacji wielomianów [6][7].

W niniejszej pracy zastosowano podejście wielomianowe w celu przekształcenia wybranych algorytmów FWT do postaci modułowej.

2. Algorytm modułowy z uporządkowaniem naturalnym

Jeden z podstawowych algorytmów szybkiej transformacji Walsha z uporządkowaniem naturalnym [8] wykorzystujący operacje motylkowe z podstawianiem można otrzymać za pomocą faktoryzacji macierzy Hadamarda w następujący sposób:

$$\underline{H}_N = \prod_{i=0}^{p-1} (\underline{I}_{2^{p-1-i}} \otimes \underline{H}_2 \otimes \underline{I}_{2^i}) \quad (1)$$

gdzie:

\underline{H}_N - macierz Hadamarda o wymiarach $N \times N$,
 \underline{I}_{2^i} - macierz jednostkowa o wymiarach $2^i \times 2^i$.

Stąd dla $N=8$ macierz \underline{H}_8 można przedstawić w postaci następującej iloczynu macierzy i odpowiadającego mu grafu przepływu sygnału (rys. 1).

$$\underline{H}_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & 1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \end{bmatrix} \quad (2)$$

Tak wyprowadzony algorytm FWT odpowiada ogólnej postaci transformacji ortogonalnej mającej strukturę drzewa (rys. 2) i opisanej następującym iloczynem macierzy [9]:

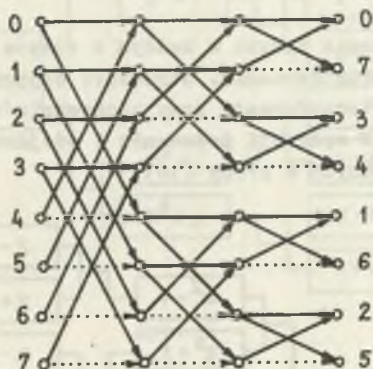
$$\begin{bmatrix} \underline{B}_1 & \vdots & \underline{O}_N \\ \dots & \vdots & \dots \\ \underline{O}_N & \vdots & \underline{B}_2 \end{bmatrix} \cdot \begin{bmatrix} a_{11}\underline{I}_N & \vdots & a_{12}\underline{I}_N \\ \dots & \vdots & \dots \\ a_{21}\underline{I}_N & \vdots & a_{22}\underline{I}_N \end{bmatrix} \quad (3)$$

dla

$$\underline{B} \underline{B}^T = \underline{L}_b, \quad \underline{A} \underline{A}^T = \underline{L}_a \quad (4)$$

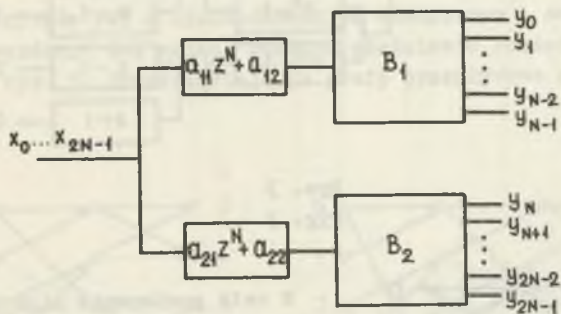
gdzie:

$$\underline{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \quad \underline{L}_a, \underline{L}_b \text{ - macierze diagonalne.}$$



Rys. 1

Fig. 1



Rys. 2

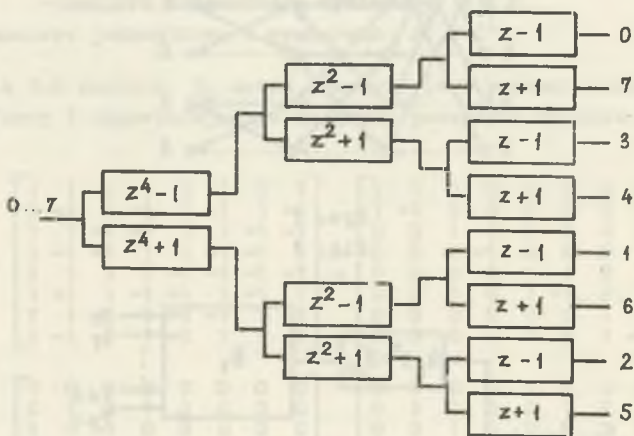
Fig. 2

W przypadku FWT z uporządkowaniem naturalnym $a_{11} = a_{12} = a_{21} = 1$, $a_{22} = -1$ i przykładowa modułowa struktura FWT dla $N=8$ ma postać przedstawioną na rys. 3.

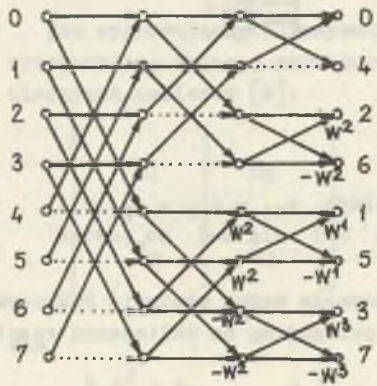
Widać stąd, że faktoryzację macierzy Hadamarda można zastąpić rekursywną faktoryzacją wielomianu $z^N - 1$, $N=2^p$, polegającą na obliczaniu reszt [6] w sposób wieloetapowy:

$$\begin{aligned}
 z^N - 1 &= (z^{N/2} - 1)(z^{N/2} + 1) \\
 z^{N/2} - 1 &= (z^{N/4} - 1)(z^{N/4} + 1) \\
 &\dots\dots\dots \\
 z^2 - 1 &= (z - 1)(z + 1)
 \end{aligned}
 \tag{5}$$

Zapewnia to rozdzielenie danych w każdym z etapów FWT i stwarza możliwość równoległego ich przetwarzania. Struktura drzewa z rys. 3 pozwala na obliczanie bieżącej transformaty przy zachowaniu złożoności obliczeniowej FWT równej $N \log_2 N$ operacji dodawania i odejmowania w odróżnieniu od N^2 operacji proponowanych w pracy [10].

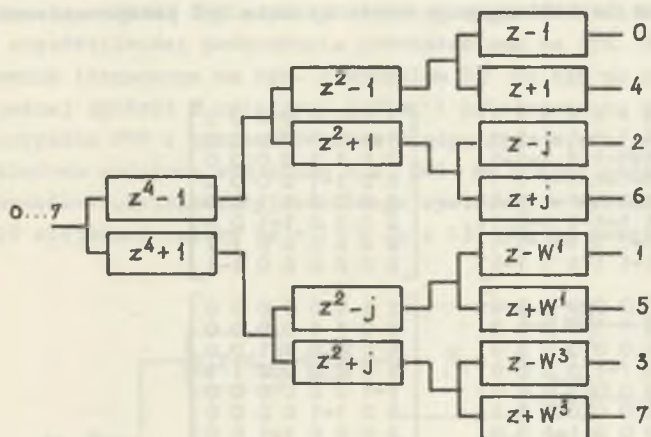


Rys. 3
Fig. 3



Rys. 4
Fig. 4

W celu porównania algorytmów FWT i FFT na rys. 4 podano graf przepływowy algorytmu FFT o takiej samej geometrii operacji motylkowych jak algorytm FWT z rys. 1, a na rys. 5 równoważną mu strukturę modułową będącą odpowiednikiem struktury z rys. 3. W odróżnieniu od algorytmów FWT algorytm FFT o takiej samej strukturze charakteryzuje się większą złożonością obliczeniową. Wymagają one $N \log_2 N$ operacji dodawania i odejmowania liczb zespolonych oraz $N/2 \log_2 N$ operacji mnożenia liczb zespolonych przez współczynniki zespolone W^k , gdzie $W = \exp(-j2\pi/N)$.

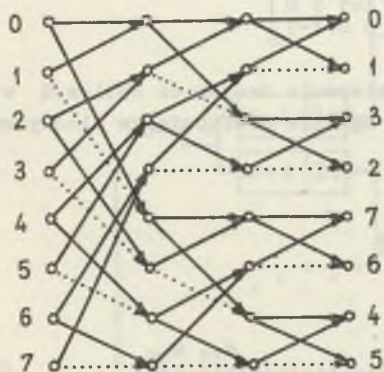


Rys. 5

Fig. 5

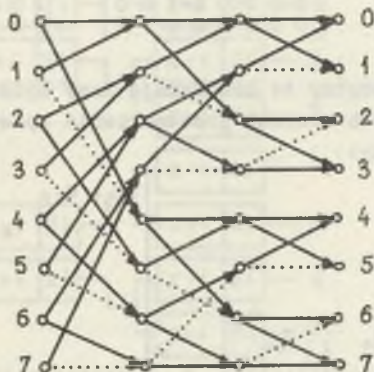
3. Algorytmy modułowe z uporządkowaniem diadycznym i sekwencyjnościowym

Spśród algorytmów FWT z uporządkowaniem diadycznym i sekwencyjnościowym [2] można wyróżnić dwa mające podobną właściwość rozdzielania danych jak algorytm z rys. 1. Przedstawiają je grafy przepływowo z rys. 6 i 7.



Rys. 6

Fig. 6

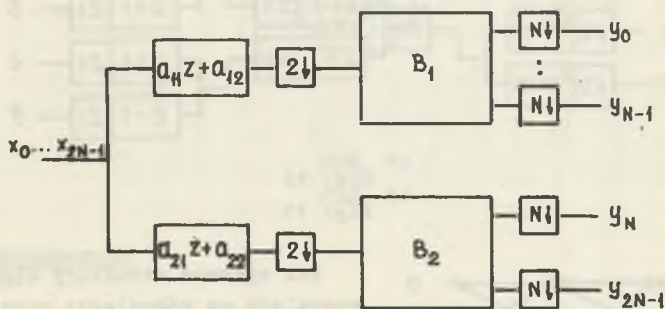


Rys. 7

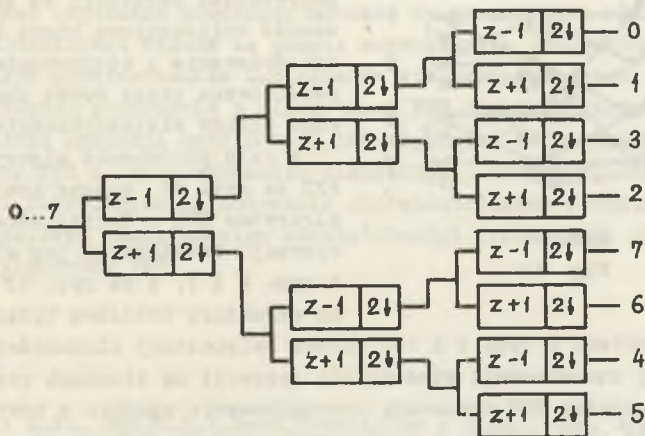
Fig. 7

Takiemu sposobowi faktoryzacji można przypisać strukturę drzewa z redukcją częstotliwości próbkowania przedstawioną na rys. 8 [9]. Redukcja próbkowania (oznaczona na rys. 8 symbolem N) polega na rejestrowaniu tylko jednej spośród N kolejnych próbek i na odrzuceniu pozostałych.

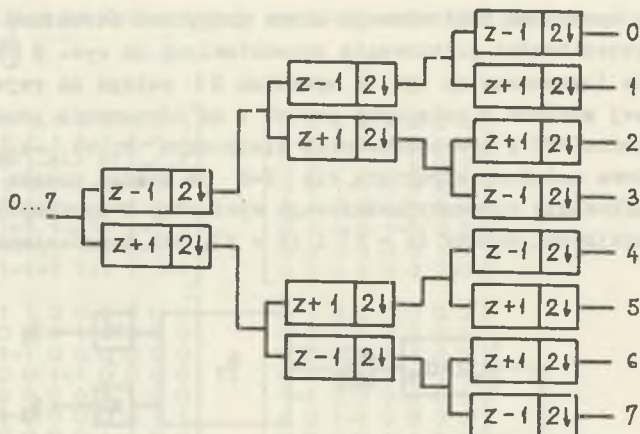
W przypadku FWT z uporządkowaniem diadycznym $a_{11}=a_{12}=a_{21}=1$, $a_{22}=-1$ i przykładowa modułowa struktura dla $N=8$ ma postać podaną na rys. 9. Dla uporządkowania sekwencyjnościowego wystarczy w wybranych gałęziach zamienić miejscami człony $(z-1)$ i $(z+1)$, jak to pokazano na rys. 10.



Rys. 8
Fig. 8

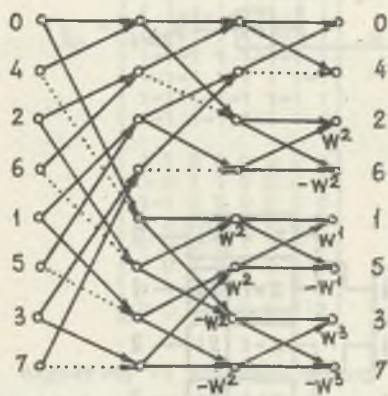


Rys. 9
Fig. 9



Rys. 10

Fig. 10



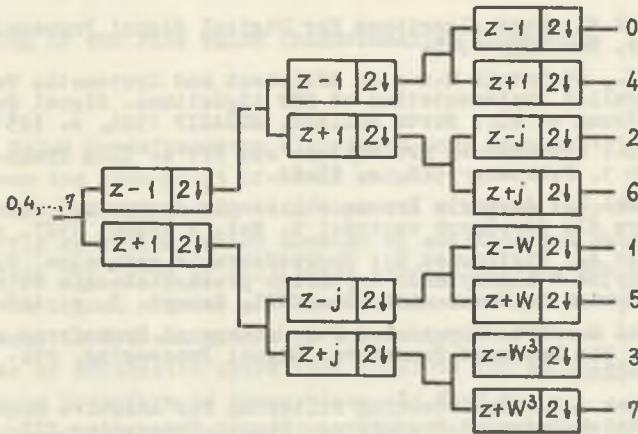
Rys. 11

Fig. 11

Tak opisane struktury algorytmów FWT pozwalają na równoległe przetwarzanie w kolejnych etapach rozdzielonych grup danych przy jednoczesnym dwukrotnym zmniejszaniu częstotliwości próbkowania w każdym etapie w porównaniu z etapem poprzednim. Struktury te zachowują złożoność obliczeniową równą $N \log_2 N$ operacji dodawania i odejmowania i mogą być realizowane przez coraz częściej stosowane układy wieloszybkociowe [11].

W celu porównania algorytmów FWT i FFT na rys. 11 podano graf przepływowy algorytmu FFT o takiej samej geometrii operacji motylkowych jak algorytmy FWT z rys. 6 i 7, a na rys. 12 równoważną mu strukturę modułową będącą odpowied-

nikiem struktury z rys. 9 i 10. Oprócz zwiększonej złożoności obliczeniowej będącej konsekwencją stosowanych operacji na liczbach zespolonych, tego typu algorytmy FFT wymuszają uporządkowanie zgodnie z odwrotną kolejnością bitów zarówno próbek wejściowych, jak i wyjściowych, co wymaga dodatkowych etapów przetwarzania i znacznie ogranicza stosowanie struktury modułowej w połączeniu z redukcją częstotliwości próbkowania.



Rys. 12

Fig. 12

4. Podsumowanie

W pracy przedstawiono modułowe struktury algorytmów FWT pozwalające na równoległe przetwarzanie danych i wielostopniowe zmniejszanie częstotliwości próbkowania. Zagadnienia te w opracowaniach i monografiach o charakterze teoretycznym dotyczących transformacji Walsh są najczęściej pomijane [12]. Opierając się na prezentowanych strukturach algorytmów FWT można budować użyteczne procedury służące do wyznaczania bieżących dyskretnych transformat Walsh za pomocą nowoczesnych procesorów sygnałowych z równoległym przetwarzaniem lub układów wieloprocesorowych. Na podstawie przeprowadzonego porównania z algorytmami FFT charakteryzującymi się taką samą geometrią operacji motylkowych jak algorytmy FWT można stwierdzić, że algorytmy FWT z uporządkowaniem diadycznym i sekwencyjnościowym mają właściwości pozwalające na budowanie skuteczniejszych struktur modułowych z wielostopniowym zmniejszaniem częstotliwości próbkowania niż odpowiadające im algorytmy FFT.

LITERATURA

- [1] Elliott D.F., Rao K.R.: Fast Transforms - Algorithms, Analyses, Applications. Academic Press, New York 1982.
- [2] Beauchamp K.G.: Applications of Walsh and Related Functions. Academic Press, London 1984.
- [3] Nussbaumer H.J.: Fast Fourier Transform and Convolution Algorithms. Springer-Verlag, Berlin 1982.

- [4] Blahut R.E.: Fast Algorithms for Digital Signal Processing. Addison-Wesley, Reading, 1985.
- [5] Pitas I., Strintzis M.G.: An Efficient and Systematic Technique for the Parallel Implementation of DFT Algorithms. Signal Processing III. I. T. Young et al., North Holland, EURASIP 1986, s. 1251-1254.
- [6] Bruun G.: z-Transform DFT Filters and FFT's. IEEE Trans., vol. ASSP-26, No 1, February 1978, s. 56-63.
- [7] Stasiński R.: Algorytm Bruuna obliczenia dyskretnej transformacji Fouriera dla dowolnych wartości N . Mat. X SPETO, 1987, s. 321-327.
- [8] Drygajko A., Pietraszek S.: Uporzadkowanie naturalne, diadyczne i sekwencyjne w algorytmach szybkiego przekształcenia Walsha. Archiwum Automatyki i Telemekhaniki, Tom XXVI, Zeszyt 3, s. 445-463.
- [9] Vetterli M.: Tree Structures for Orthogonal Transforms and Application to the Hadamard Transform. Signal Processing, vol. 5, 1983, s. 473-484.
- [10] Karbowski A.E.: Goal Seeking Filtering for Adaptive Signal Analysis Using Walsh-Hadamard Transforms. Signal Processing III. I. T. Young et al., North Holland, EURASIP 1986, s. 69-72.
- [11] Crochiere R.E., Rabiner L.R.: Multirate Digital Signal Processing. Prentice-Hall, Englewood Cliffs, New Jersey 1983.
- [12] Gołubow B.I., Jefimow A.W., Skworcow W.A.: Riady i preobrazowanija Walsha. Nauka, Moskwa 1987.

Recenzent: doc. dr inż. Zdzisław Trzaska

Wpłynęło do redakcji dnia 20 maja 1988 r.

МОДУЛЬНАЯ ФОРМА АЛГОРИТМОВ БЫСТРОГО ПРЕОБРАЗОВАНИЯ УОЛША

Р е з ю м е

В работе проанализованы алгоритмы быстрого преобразования Уолша с точки зрения их модульности. Представлены эквивалентные структуры деревьев, которые позволяют на текущее параллельное выполнение основных мотыльковых операций, состоящих из операций сложения и вычитания. Рассмотрены два варианта многоступенчатого преобразования. Первый - использующий задержки временных выборок и второй - основанный на перемене частоты дискретизации в очередных этапах обработки. Количество вычислений для получения текущего преобразования для точек оказывается пропорциональным.

Представлены структуры для натурального диадного и секвенционного упорядочения в алгоритмах быстрого преобразования Уолша и сравнение их с соответствующими структурами алгоритмов быстрого преобразования Фурье.

A MODULAR FORM OF THE FAST WALSH TRANSFORMATION ALGORITHMS

S u m m a r y

The fast Walsh transformation algorithms have been investigated in the paper from the view-point of their modularity. Three structures corresponding with them and allowing to perform simultaneous butterfly operations which consist of addition and substraction have been presented. Two forms of multistage processing: the one using time delays between signal samples and another one based on the change of sampling frequency in sequent stages of processing, have been considered.

The number of arithmetic operations required for calculation of the N -point running transform is proportional to $N \log_2 N$.

The structures for natural, dyadic and sequence order in the fast Walsh transformation algorithms have been given and compared with the fast Fourier transformation algorithms structures corresponding with them.