

# Cyfrowe przetwarzanie sygnału

## Zadanie 4

### **Przekształcenie Fouriera, Walsha-Hadamarda, kosinusowe i falkowe, szybkie algorytmy.**

Celem ćwiczenia jest zapoznanie się z operacjami transformacji sygnałów dyskretnych przy użyciu wybranych metod.

#### ***Zakres funkcjonalności aplikacji***

Do programu przygotowywanego w ramach dotychczasowych ćwiczeń należy dodać możliwość odczytu i zapisu sygnałów o wartościach zespolonych. Należy również umożliwić rysowanie wykresów sygnałów dyskretnych o wartościach zespolonych w postaci dwóch wykresów o wspólnej dziedzinie ułożonych jeden nad drugim. Można przyjąć, że sygnały zespolone będą tylko wynikiem transformacji Fouriera, a co za tym idzie będą prezentować funkcje w dziedzinie częstotliwości. Dostępne mają być dwa tryby prezentacji wykresu:

- (W1) – górny wykres prezentuje część rzeczywistą amplitudy w funkcji częstotliwości, a wykres dolny część urojoną;
- (W2) – górny wykres prezentuje moduł liczby zespolonej, a dolny argument liczby w funkcji częstotliwości.

Należy zaimplementować następujące transformacje sygnałów:

- (F1) – dyskretna transformacja Fouriera – algorytm z definicji oraz szybka transformacja Fouriera z decymacją w dziedzinie czasu (DIT FFT);
- (F2) – dyskretna transformacja Fouriera – algorytm z definicji oraz szybka transformacja Fouriera z decymacją w dziedzinie częstotliwości (DIF FFT);
- (T1) – transformacja kosinusowa typu drugiego (DCT II) oraz szybka transformacja kosinusowa (FCT II);
- (T2) – transformacja Walsha-Hadamarda oraz szybka transformacja Walsha-Hadamarda;
- (T3) – transformacja falkowa (jeden poziom), warianty:
  - (DB4), (DB6), (DB8).

W implementacji należy uwzględnić możliwość transformacji sygnałów o liczbie próbek  $2^n$  dla  $n$  całkowitego w zakresie od  $n=1$  do  $n=10$  (próbki o długości od 2 do 1024).

Po wykonaniu transformacji program ma podać czas wykonywania operacji (porównanie szybkości działania algorytmów).

Wykorzystując mechanizmy z zadania pierwszego utworzyć i zapisać do pliku sygnał dany następującym równaniem:

- (S1) -  $S(t) = 2 \sin\left(\frac{2\pi}{2}t + \frac{\pi}{2}\right) + 5 \sin\left(\frac{2\pi}{0,5}t + \frac{\pi}{2}\right) f_{pr} = 16$
- (S2) -  $S(t) = 2 \sin\left(\frac{2\pi}{2}t\right) + \sin\left(\frac{2\pi}{1}t\right) + 5 \sin\left(\frac{2\pi}{0,5}t\right) f_{pr} = 16$
- (S3) -  $S(t) = 5 \sin\left(\frac{2\pi}{2}t\right) + \sin\left(\frac{2\pi}{0,25}t\right) f_{pr} = 16$

W sprawozdaniu należy umieścić wykresy prezentujące wygenerowane sygnały oraz ich transformaty. Należy także określić jak zmienia się szybkość wykonania obliczeń w zależności od rodzaju zastosowanego algorytmu (dla przypadków kiedy istnieje szybka transformacja).

### **Dyskretna transformacja Fouriera (DFT)**

Dyskretna transformacja Fouriera jest przekształceniem macierzowym o postaci:

$$X(m) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) e^{-j \frac{2\pi m}{N} n} \quad 0 \leq m \leq N-1 \quad (\text{F-1})$$

Przekształcenie do niego odwrotne wyraża się zależnością:

$$x(n) = \sum_{m=0}^{N-1} X(m) e^{j \frac{2\pi m}{N} n} \quad 0 \leq n \leq N-1 \quad (\text{F-2})$$

Oznaczenia w powyższych wzorach należy interpretować następująco:

$x(n)$  – wektor próbek sygnału, kolejne próbki są oddalone od siebie o czas  $\Delta t = 1/f_{pr}$  (1/częstotliwość próbkowania).

$X(m)$  – wektor będący wynikiem transformacji sygnału  $x(n)$ . Kolejne wartości oznaczają udział składowej o częstotliwości  $m \cdot f_0$  w sygnale próbkowanym. Częstotliwość  $f_0$  równa:

$$f_0 = \frac{1}{T} = \frac{1}{\Delta t N} = \frac{f_{pr}}{N} \quad (F-3)$$

Wyrażenie  $e^{\pm j \frac{2\pi m}{N} n}$  nazywamy jądrem transformacji (prostej ze znakiem „-” i odwrotnej ze znakiem „+”). Często dla poprawy czytelności stosowany jest zapis:

$$W_N = e^{j \frac{2\pi}{N}} \quad (F-4)$$

Wtedy równania transformacji prostej i odwrotnej mają postać:

$$X(m) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) W_N^{-mn}, \quad x(n) = \sum_{m=0}^{N-1} X(m) W_N^{mn} \quad 0 \leq m, n \leq N-1 \quad (F-5)$$

Postać macierzowa:

$$\begin{bmatrix} X_0 \\ X_1 \\ X_2 \\ \dots \\ X_{N-1} \end{bmatrix} = \frac{1}{N} \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & W_N^{-1} & W_N^{-2} & \dots & W_N^{-(N-1)} \\ 1 & W_N^{-2} & W_N^{-4} & \dots & W_N^{-2(N-1)} \\ \dots & \dots & \dots & \dots & \dots \\ 1 & W_N^{-(N-1)} & W_N^{-2(N-1)} & \dots & W_N^{-(N-1)(N-1)} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \dots \\ x_{N-1} \end{bmatrix} \quad (F-6)$$

## ***Szybka transformacja Fouriera (FFT)***

Przedstawiona powyżej dyskretna transformacja Fouriera wymaga mnożenia macierzy o rozmiarze  $N \times N$  przez wektor długości  $N$ . Algorytm taki wymaga  $N^2$  mnożeń liczb zespolonych. Ilość tych operacji rośnie wprost proporcjonalnie do kwadratu długości wektora próbki danych, a więc jest to algorytm o złożoności  $O(N^2)$ . Wykorzystanie pewnych charakterystycznych cech transformacji pozwala budować znacznie wydajniejsze algorytmy, których złożoność obliczeniowa wynosi  $O(N \log_2 N)$ .

### **Decymacja w dziedzinie czasu**

Równanie transformacji można zapisać rozdzielając próbki parzyste i nieparzyste do osobnych wektorów:

$$X(m) = \frac{1}{N} \left( \sum_{n=0}^{N/2-1} x(2n) W_N^{-m(2n)} + \sum_{n=0}^{N/2-1} x(2n+1) W_N^{-m(2n+1)} \right) \quad (F-7)$$

Ze względu na fakt, że podział dotyczy próbek wejściowych, zależnych od czasu, podział ten jest nazwany decymacją w dziedzinie czasu.

Po wyciągnięciu czynnika  $W_N^{-m}$  przed znak sumy w drugim członie równanie ma postać:

$$X(m) = \frac{1}{N} \left( \sum_{n=0}^{N/2-1} x(2n) W_N^{-m(2n)} + W_N^{-m} \sum_{n=0}^{N/2-1} x(2n+1) W_N^{-m(2n)} \right) \quad (F-8)$$

Oba wyrażenia określające sumowanie mają taką samą postać z tym, że lewe dotyczy wyrazów nieparzystych a prawe parzystych wejściowego wektora  $x(n)$ . Dodatkowo zachodzi równość:

$$W_N^{-m(2n)} = \left( e^{j \frac{2\pi}{N}} \right)^{-2mn} = e^{j \frac{2\pi}{N}(-2mn)} = e^{j \frac{2\pi}{N/2}(-mn)} = W_{N/2}^{-mn} \quad (F-9)$$

Zatem poprzednie wyrażenie można zapisać:

$$X(m) = \frac{1}{N} \left( \sum_{n=0}^{N/2-1} x(2n) W_{N/2}^{-mn} + W_N^{-m} \sum_{n=0}^{N/2-1} x(2n+1) W_{N/2}^{-mn} \right) \quad (F-10)$$

Wyrażenia pod znakiem sumy są analogiczne do wejściowej postaci, przy czym sumowanie jest w zakresie do  $N/2$  (suma dotyczy połowy próbek). Współczynnik  $m$  ma jednak wartości od 0 do  $N-1$ . Ponieważ mamy:

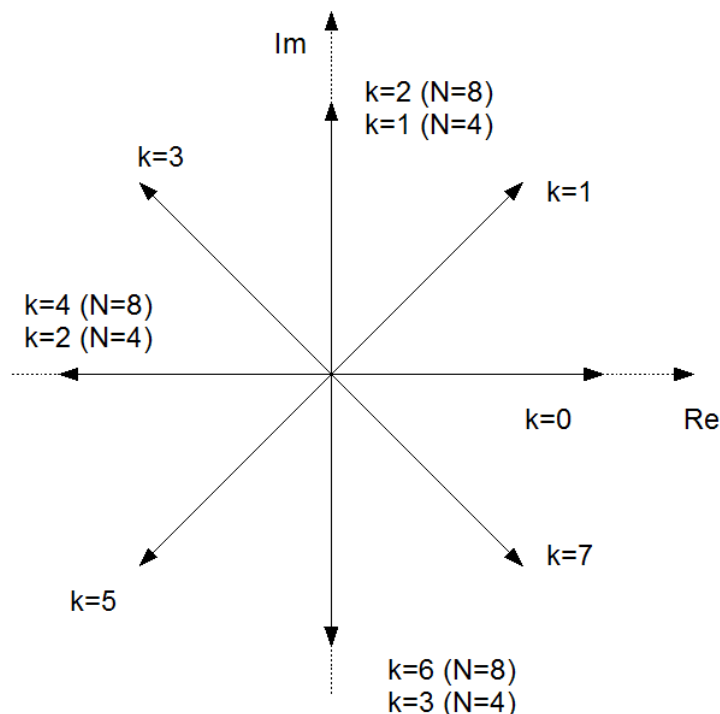
$$\begin{aligned} W_{N/2}^{-(m+N/2)n} &= W_{N/2}^{-mn} W_{N/2}^{-(N/2)n} = W_{N/2}^{-mn} e^{-j \frac{2\pi}{N/2}(N/2)n} = W_{N/2}^{-mn} e^{-j 2\pi n} = \\ &= W_{N/2}^{-mn} (\cos(2\pi n) - j \sin(2\pi n)) = W_{N/2}^{-mn} (1 - j*0) = W_{N/2}^{-mn} \end{aligned} \quad (F-11)$$

Oznacza to, że druga połowa wektora jest powtórzeniem pierwszej połowy. Dzięki temu zamiast wykonywać operację sumowania  $N$ -krotnie wystarczy wykonać ją  $N/2$  razy, a następnie skopiować wyniki do drugiej połowy wektora. To samo należy zastosować do wektora obliczonego z próbek nieparzystych. Daje to oszczędność obliczeń, bo początkowo wymagane było mnożenie macierzy  $N \times N$ , a obecnie mnożone są dwie macierze o wymiarach  $N/2 \times N/2$ , czyli jest to  $2 \cdot (N/2)^2 = N^2/2$  operacji. Pozostaje jeszcze

wykonanie  $N$  mnożeń przy składaniu obu wektorów, ale koszt tej operacji jest mniejszy, niż uzyskane dotychczas oszczędności.

Dodatkowo operację taką można zastosować wobec każdego z wektorów próbek parzystych i nieparzystych aż do uzyskania transformacji dwupunktowej. Zastosowanie rekurencji w tym przypadku jest jak najbardziej na miejscu. Wadą jest jednak to, że dla każdego zagnieżdżenia trzeba alokować nowe obszary pamięci na tablice. Dlatego też zostały opracowane algorytmy *in situ*, które operują na stałej ilości pamięci zamieniając wartości tablicy w kolejnych etapach. Zasada działania opiera się na tym, że najpierw wykonywane są obliczenia na najkrótszych wektorach (dwuelementowych). Następnie wyniki tych operacji składane są w wektory czteroelementowe, te w ośmio itd.

Jak poprzednio wykazano współczynnik  $W_N^k$  jest funkcją okresową względem wykładnika o okresie  $N$ . Ponieważ są to liczby zespolone kolejne wartości tego współczynnika dla wartości  $N=8$  i  $N=4$  można przedstawić na płaszczyźnie zespolonej jak poniżej:



Wartości tego współczynnika zestawiono w tabeli:

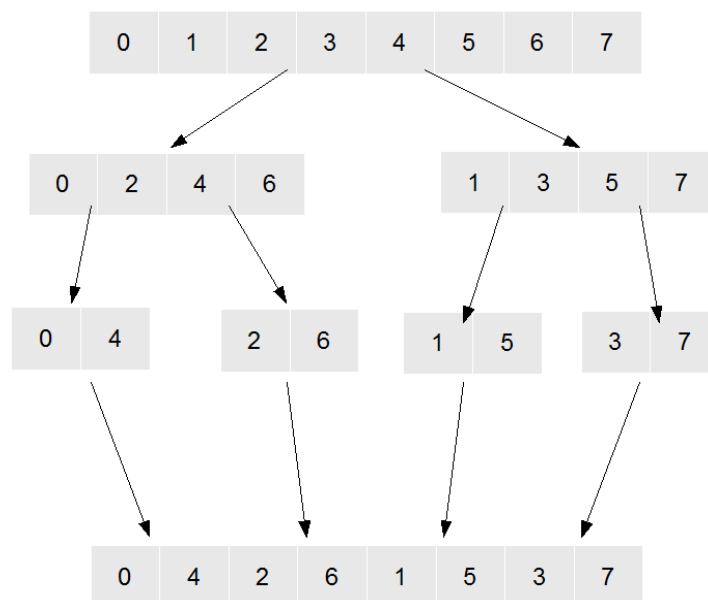
$k$	0	1	2	3	4	5	6	7
$W_N^k$	$1+0j$	$0,7071 + j0,7071$	$0+j$	$-0,7071 + j0,7071$	$-1-0j$	$-0,7071 - j0,7071$	$-0-j$	$0,7071 - j0,7071$

Dwie prawidłowości, jakie można zaobserwować na podstawie wzorów, rysunku i tabelki:

- wartości współczynników dla wektora długości  $N/2$  są równe parzystym współczynnikom obliczonym dla wektora o długości  $N$ ;
- wartości współczynnika w drugiej połowie tabeli są przeciwne do wartości w pierwszej połowie, czyli  $W_N^{(k+N/2)} = -W_N^{(k)}$ .

Dzięki temu zamiast obliczać macierz współczynników o rozmiarze  $N \times N$  wystarczy obliczyć wektor długości  $N/2$ .

Pierwszym krokiem szybkiego algorytmu działającego w miejscu jest przestawienie kolejności próbek. Wynika to z opisanej wcześniej konieczności rozdzielenia elementów parzystych i nieparzystych. Dla wektora o długości 8 próbek nowa kolejność jest ustalana następująco:



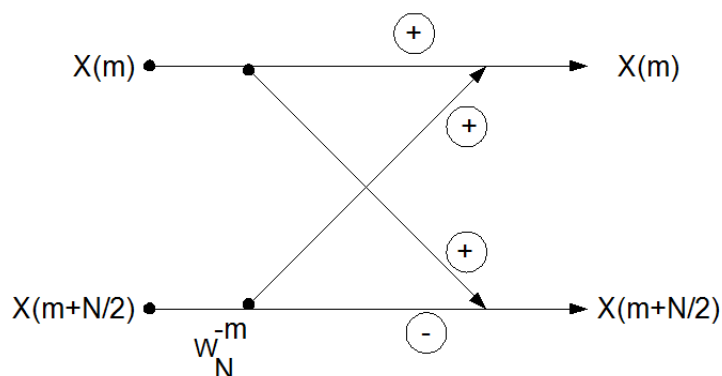
Ogólny algorytm przestawiania próbek w wektorze oparty jest o zmianę kolejności bitów w indeksie próbki (w zapisie dwójkowym):

Numer próbki wejściowej (dziesiętnie)	0	1	2	3	4	5	6	7
Numer próbki wejściowej (dwójkowo)	000	001	010	011	100	101	110	111
Numer próbki wyjściowej (dwójkowo)	000	100	010	110	001	101	011	111
Numer próbki wyjściowej (dziesiętnie)	0	4	2	6	1	5	3	7

Wektor z przestawionymi próbkami można traktować jako wektor zawierający  $N$  jednopunktowych transformat. Z tego powodu można zastosować wzór F-10 i złożyć je w  $N/2$  transformat dwupunktowych, te następnie w  $N/4$  czteropunktowych i w ten sposób uzyskać jedną transformatę  $N$ -punktową.

W każdym etapie składania współczynniki  $W_N^m$  można pobierać z wyliczonego wcześniej wektora długości  $N/2$ . Należy jednak pamiętać, że długość tego wektora jest stała i jest to połowa długości transformaty. W zapisie  $W_N^m$   $N$  zmienia na każdym etapie, podwaja się ono w stosunku do wartości z poprzedniego etapu.

Wszystkie etapy obliczeń wygodnie jest zapisywać przy pomocy struktury zwanej motylkiem. Struktura ta wygląda następująco:

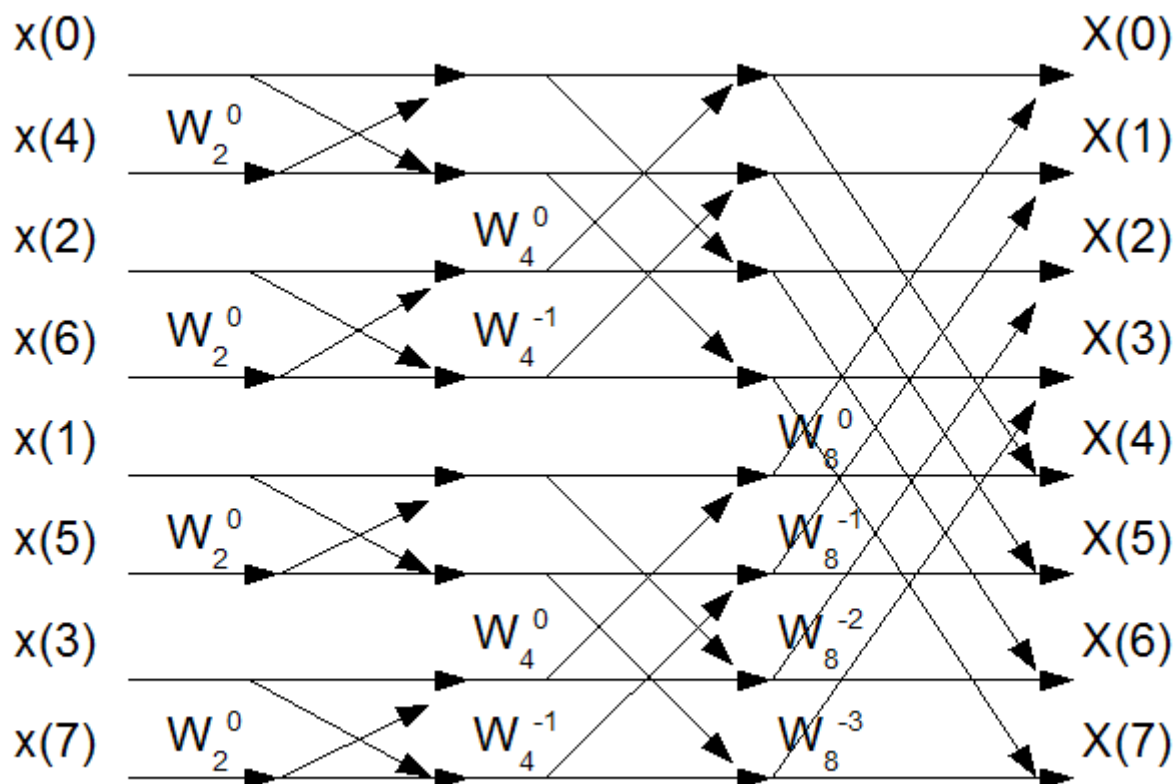


Przedstawione operacje należy interpretować następująco: wartość  $X(m+N/2)$  należy pomnożyć przez odpowiedni współczynnik  $W_N^{-m}$  odczytany z wektora. Wyniki po prawej stronie uzyskuje się jako suma wartości ze strony lewej ( $X(m)$ ) oraz jako różnica tych wartości ( $X(m+N/2)$ ). Znak minus wynika z faktu, że w drugiej połowie wektora współczynników są wartości przeciwne do tych z pierwszej. W zapisie macierzowym motylkowi odpowiada zapis:

$$\begin{bmatrix} X(m) \\ X(m+N/2) \end{bmatrix} = \begin{bmatrix} 1 & W_N^{-m} \\ 1 & -W_N^{-m} \end{bmatrix} \begin{bmatrix} X(m) \\ X(m+N/2) \end{bmatrix} \text{ dla } m=0, 1, \dots, N/2-1 \quad (\text{F-12})$$



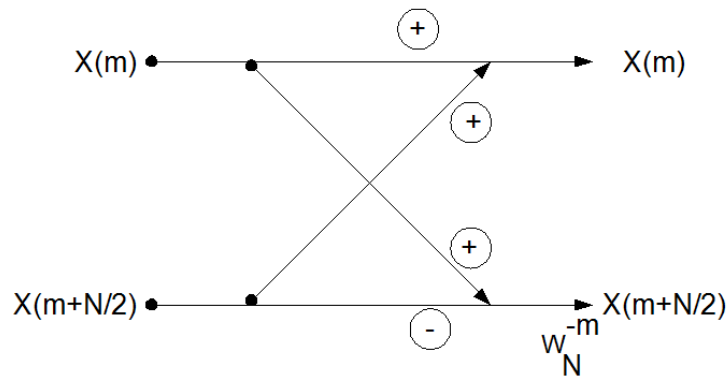
Szybka transformacja wektora 8 próbek jest przedstawiona na rysunku poniżej. Pokazane jest które współczynniki są użyte do mnożenia składowej nieparzystej na poszczególnych etapach. Na rysunku pominięte są oznaczenia operacji dodawania i odejmowania, ale są one takie same, jak na rysunku prezentującym strukturę motylka.



Transformacja odwrotna przebiega w sposób analogiczny, jedynie wektor współczynników należy utworzyć bez znaku minus przed wykładnikiem oraz wynik nie jest dzielony przez  $N$ .

### **Decymacja w dziedzinie częstotliwości**

Alternatywną metodą tworzenia szybkiego algorytmu jest decymacja w dziedzinie częstotliwości. Algorytm jest podobny do przedstawionego powyżej, ale jego wyprowadzenie rozpoczyna się od podziału wyniku transformacji na próbki parzyste i nieparzyste. Zasadnicza różnica polega na tym, że w pojedynczym motylku najpierw wykonywane są operacje dodawania i odejmowania, a dopiero potem wynik jest mnożony przez odpowiedni współczynnik transformacji.



### **Transformacja kosinusowa (DCT)**

Istnieje kilka typów przekształcenia kosinusowego, które różnią się między sobą parametrami jądra przekształcenia. Wspólną cechą jest wykorzystanie funkcji kosinus o różnych fazach i okresach.

DCT II jest zdefiniowane w sposób następujący:

$$X_m = c(m) \sum_{n=0}^{N-1} x(n) \cos\left(\frac{\pi(2n+1)m}{2N}\right), \quad 0 \leq m, n \leq N-1 \quad (\text{C-1})$$

Przekształcenie odwrotne dane jest wzorem:

$$x(n) = \sum_{m=0}^{N-1} c(m) X(m) \cos\left(\frac{\pi(2n+1)m}{2N}\right), \quad 0 \leq m, n \leq N-1 \quad (\text{C-2})$$

$$c(0) = \sqrt{\frac{1}{N}}, \quad c(m) = \sqrt{\frac{2}{N}} \quad \text{dla } 0 \leq m \leq N-1 \quad (\text{C-3})$$

### **Szybka transformacja kosinusowa (FCT)**

Istnieje wiele metod optymalizacji algorytmu DCT. Jedną z nich jest przekształcenie równania DCT do postaci pozwalającej wyodrębnić z niego człon stanowiący DFT. To z kolei pozwala na zastosowanie szybkich algorytmów opracowanych dla dyskretnej transformacji Fouriera.

Mając dany sygnał dyskretny jako funkcję  $x(n)$  dla  $0 \leq n \leq N-1$  można zdefiniować sygnał  $y(n)$  następująco:

$$y(n) = x(2n) \quad \text{ i } \quad y(N-1-n) = x(2n+1) \quad \text{ dla } \quad 0 \leq n \leq \frac{N}{2}-1 \quad (\text{C-4})$$

Nowy sygnał  $y(n)$  jest w istocie sygnałem  $x(n)$  o zmienionej kolejności próbek: w pierwszej połowie znajdują się próbki parzyste, a w drugiej próbki nieparzyste, ale w odwróconej kolejności. Przykładowo:

$$x(n): 0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7$$

$$y(n): 0 \ 2 \ 4 \ 6 \ 7 \ 5 \ 3 \ 1$$

Transformatę sygnału  $x(n)$  można obliczyć korzystając z następującej zależności:

$$X(m) = Re[c(m) e^{-j\pi m/2N} DFT_N(y(n))] \quad (C-5)$$

Przekształcenie odwrotne dane jest równaniem:

$$y(n) = Re[IDFT_N\{c(m) e^{j\pi m/2N} X(m)\}] \quad (C-6)$$

Sygnał  $x(n)$  należy odtworzyć poprzez zmianę kolejności próbek w sygnale  $y(n)$  zgodnie z podaną wcześniej zależnością:

$$y(n): 0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7$$

$$x(n): 0 \ 7 \ 1 \ 6 \ 2 \ 5 \ 3 \ 4$$

### **Transformacja Walsh-Hadamarda**

Transformacja ta jest określona przekształceniem macierzowym następująco:

$$X = H_m x \quad (WH-1)$$

Przekształcenie odwrotne ma postać:

$$x = H_m X \quad (WH-2)$$

W powyższych równaniach  $x$  oznacza wektor próbek wejściowych,  $X$  jest wektorem próbek po transformacji natomiast  $H_m$  jest macierzą Hadamarda zdefiniowaną rekurencyjnie w następujący sposób:

$$H_0 = 1$$

$$H_m = \frac{1}{\sqrt{2}} \begin{bmatrix} H_{m-1} & H_{m-1} \\ H_{m-1} & -H_{m-1} \end{bmatrix} \quad (WH-3)$$

Ze względu na taki sposób definicji macierzy ma ona rozmiar  $2^m \times 2^m$ .

### **Szybki algorytm transformacji Walsh-Hadamarda**

Budowa algorytmu opiera się na zasadzie podziału wektora wynikowego na połowy i osobnym obliczaniu wartości dla pierwszej i drugiej połowy. Przyjmując jako  $m$  indeks

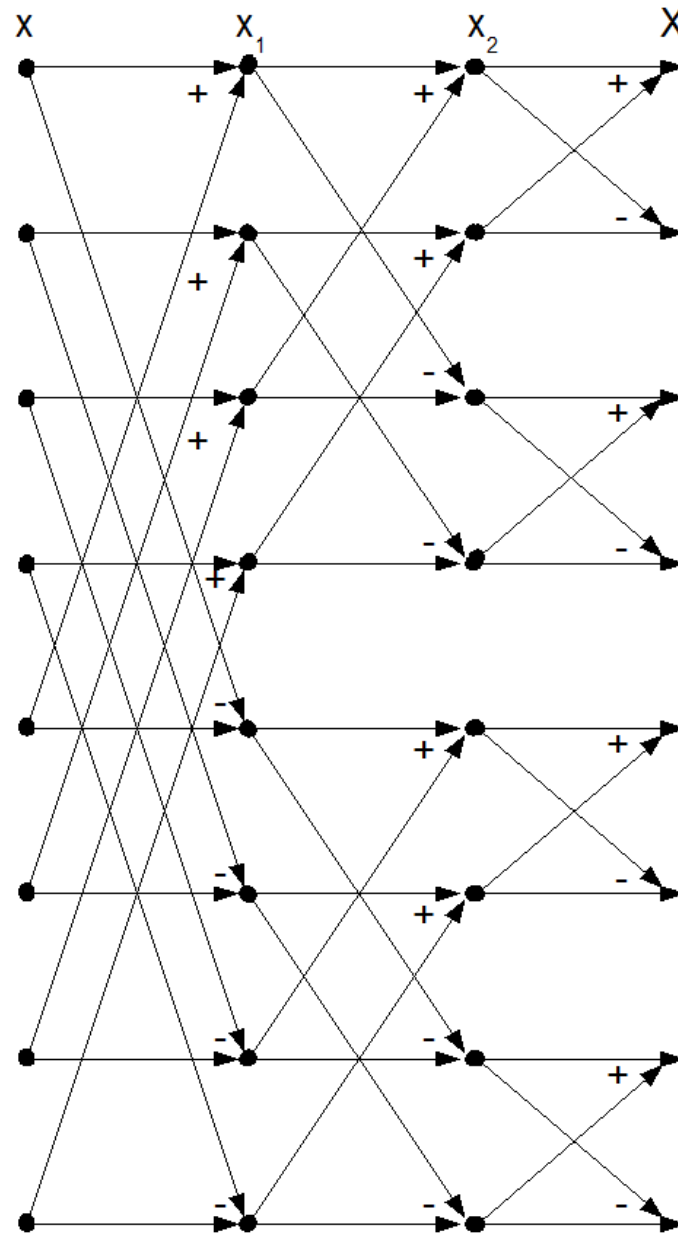
macierzy długość wektora próbek będzie wynosiła  $N=2^m$ . Czynniki skalujące  $\frac{1}{\sqrt{2}}$  dla uproszczenia został pominięty, ale należy pamiętać, aby na końcu pomnożyć wynikowy wektor przez odpowiednią stałą.

$$\begin{bmatrix} X(0) \\ \dots \\ X(N/2-1) \\ X(N/2) \\ \dots \\ X(N-1) \end{bmatrix} = H_m \begin{bmatrix} x(0) \\ \dots \\ x(N/2-1) \\ x(N/2) \\ \dots \\ x(N-1) \end{bmatrix} = \begin{bmatrix} H_{m-1} & H_{m-1} \\ H_{m-1} & -H_{m-1} \end{bmatrix} \begin{bmatrix} x(0) \\ \dots \\ x(N/2-1) \\ x(N/2) \\ \dots \\ x(N-1) \end{bmatrix} \quad (\text{WH-4})$$

$$\begin{aligned} \begin{bmatrix} X(0) \\ \dots \\ X(N/2-1) \end{bmatrix} &= \begin{bmatrix} H_{m-1} & H_{m-1} \end{bmatrix} \begin{bmatrix} x(0) \\ \dots \\ x(N/2-1) \\ x(N/2) \\ \dots \\ x(N-1) \end{bmatrix} = \\ &= \begin{bmatrix} H_{m-1} \end{bmatrix} \begin{bmatrix} x(0) \\ \dots \\ x(N/2-1) \end{bmatrix} + \begin{bmatrix} H_{m-1} \end{bmatrix} \begin{bmatrix} x(N/2) \\ \dots \\ x(N-1) \end{bmatrix} = \begin{bmatrix} H_{m-1} \end{bmatrix} \begin{bmatrix} x(0)+x(N/2) \\ \dots \\ x(N/2-1)+x(N-1) \end{bmatrix} \end{aligned} \quad (\text{WH-5})$$

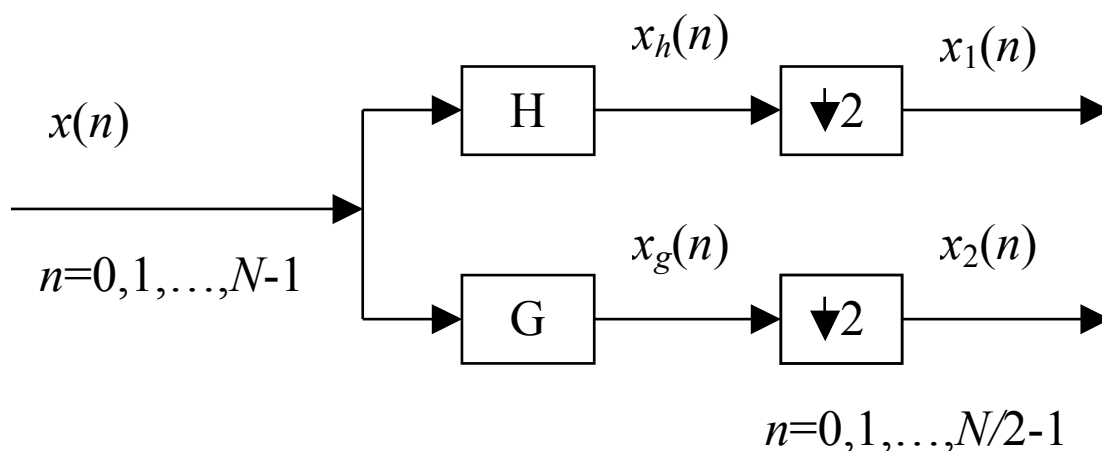
$$\begin{aligned} \begin{bmatrix} X(N/2) \\ \dots \\ X(N-1) \end{bmatrix} &= \begin{bmatrix} H_{m-1} & -H_{m-1} \end{bmatrix} \begin{bmatrix} x(0) \\ \dots \\ x(N/2-1) \\ x(N/2) \\ \dots \\ x(N-1) \end{bmatrix} = \\ &= \begin{bmatrix} H_{m-1} \end{bmatrix} \begin{bmatrix} x(0) \\ \dots \\ x(N/2-1) \end{bmatrix} - \begin{bmatrix} H_{m-1} \end{bmatrix} \begin{bmatrix} x(N/2) \\ \dots \\ x(N-1) \end{bmatrix} = \begin{bmatrix} H_{m-1} \end{bmatrix} \begin{bmatrix} x(0)-x(N/2) \\ \dots \\ x(N/2-1)-x(N-1) \end{bmatrix} \end{aligned}$$

Do obliczenia pierwszej połowy wynikowej transformaty potrzebny jest wektor długości  $N/2$  obliczony jako suma elementów wektora wejściowego oddalonych od siebie o  $N/2$ . Natomiast druga połowa jest liczona z wektora będącego różnicą elementów wektora wejściowego. Proces transformacji wektora o długości  $N=8$  został przedstawiony graficznie za pomocą schematu motylkowego na poniższym rysunku.



## Transformacja falkowa (przekształcenie falkowe - PF)

Jeden etap prostego PF opisuje schemat:



Bloki  $H$  i  $G$  są to odpowiednio filtry dolnoprzepustowy i górnoprzepustowy o odpowiedziach impulsowych.

$$H = [h_0, h_1, \dots, h_{K-1}] \quad G = [g_0, g_1, \dots, g_{K-1}] \quad (\text{TF-1})$$

Bloki te realizują operację splotu sygnału z odpowiednio filtrami  $H$  i  $G$ .

$$x_h(n) = \sum_{k=0}^{K-1} h_k x(n-k) \quad x_g(n) = \sum_{k=0}^{K-1} g_k x(n-k) \quad (\text{TF-2})$$

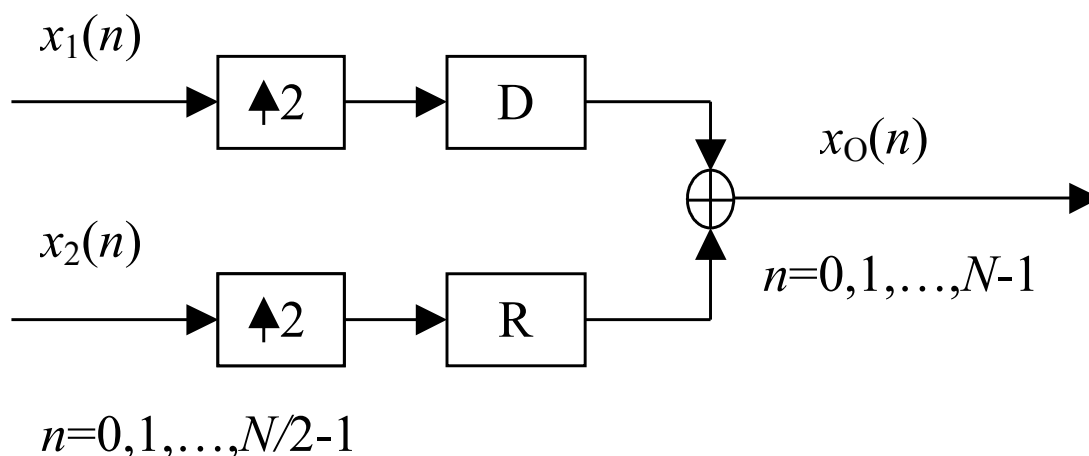
W blokach „ $\downarrow 2$ ” wyrzucamy co drugą próbkę o indeksach nieparzystych (po filtrze  $H$ ) i parzystych (po filtrze  $G$ ).

Jeden etap PF dzieli sygnał  $N$ -punktowy  $x(n)$  na dwa  $N/2$  punktowe sygnały wyjściowe  $x_1(n)$  i  $x_2(n)$ .

Dla uzyskania całościowego PF stosujemy dwa schematy kontynuacji:

- *równoległy*, gdy etapy podziału stosowane są rekurencyjnie do każdego z sygnałów wyjściowych;
- *piramidalny*, gdy rekurencyjnie stosujemy etap podziału tylko do podziału jednego z sygnałów wyjściowych.

Jeden etap odwrotnego PF opisuje schemat.



Tutaj w blokach „ $\uparrow 2$ ” wykonujemy wstawianie zerowych wartości w miejsca próbek wyrzuconych w podczas prostego PF. Bloki D i R są to filtry o odpowiedziach impulsowych

$$D = [d_0, d_1, \dots, d_{K-1}], \quad R = [r_0, r_1, \dots, r_{K-1}].$$

Kluczowym momentem syntezy szybkich PF jest dobór odpowiedzi impulsowych H, G, D i R tak, żeby spełniony był warunek idealnej rekonstrukcji opisany równaniem:

$$x \equiv P^{-1} P x \tag{TF-3}$$

gdzie  $x$  jest to wektor z elementów sygnału wejściowego  $x(n)$ ,  $P$  i  $P^{-1}$  są odpowiednio macierzami prostego i odwrotnego PF o rozmiarach  $N \times N$ .

Macierz  $P$  tworzona jest poprzez cykliczne  $N$ -punktowe przesunięcie w prawo o dwa elementy pary odpowiedzi impulsowych  $H$  oraz  $G$  i ma strukturę rzadką.

$$\mathbf{P} = \begin{bmatrix}
 h_0 & h_1 & h_2 & h_3 & \dots & h_{K-2} & h_{K-1} & & & \\
 g_0 & g_1 & g_2 & g_3 & \dots & g_{K-2} & g_{K-1} & & & \\
 & & h_0 & h_1 & \dots & h_{K-4} & h_{K-3} & h_{K-2} & h_{K-1} & \\
 & & g_0 & g_1 & \dots & g_{K-4} & g_{K-3} & g_{K-2} & g_{K-1} & \\
 & & & & \dots & & & & & \\
 & & & & & h_0 & h_1 & h_2 & h_3 & \dots & h_{K-2} & h_{K-1} \\
 & & & & & g_0 & g_1 & g_2 & g_3 & \dots & g_{K-2} & g_{K-1} \\
 h_{K-2} & h_{K-1} & & & & & & h_0 & h_1 & \dots & h_{K-4} & h_{K-3} \\
 g_{K-2} & g_{K-1} & & & & & & g_0 & g_1 & \dots & g_{K-4} & g_{K-3} \\
 & & & & \dots & & & & & & & \\
 h_2 & h_3 & h_4 & h_5 & \dots & h_{K-2} & h_{K-1} & & & & h_0 & h_1 \\
 g_2 & g_3 & g_4 & g_5 & \dots & g_{K-2} & g_{K-1} & & & & g_0 & g_1
 \end{bmatrix} \quad (\text{TF-4})$$

gdzie pokazane tylko niezerowe elementy macierzy.

Składającą się z elementów filtrów  $D$  i  $R$  macierz  $P^{-1}$  jest odwrotną do macierzy  $P$ . O szybkości przekształcenia decyduje rzadkość macierzy. Oznacza to, że tylko para macierzy  $P$  i  $P^{-1}$  o rzadkiej strukturze tworzy szybkie przekształcenia falkowe.

Najprostsze podejście do syntezy pary wyżej wymienionych macierzy polega na znalezieniu ortogonalnej macierzy  $P$ . Wtedy zachodzi równość  $P^{-1}=P^T$ , która gwarantuje rzadkość macierzy  $P^{-1}$ .

Warunki ortogonalności macierzy  $P$  opisują równania:

$$\sum_{k=0}^{K-1} h_k^2 = 1, \quad \sum_{k=0}^{K-1} h_k h_{k+2p} = 0, \quad p = K/2 - 1, K/2 - 2, \dots, 1, \quad (\text{TF-5})$$

$$g_k = \pm (-1)^k h_{N-1-k}, \quad k = 0, 1, \dots, K-1$$



Jak widać z równań, mamy do znalezienia  $K$  wartości odpowiedzi impulsowej  $h_k$  mając  $K/2$  równań. Niewystarczające  $K/2$  równań możemy utworzyć w dowolny sposób, zakładając pożądane właściwości przekształcenia falkowego. Jeśli chcemy np. uzyskać ortogonalne przekształcenie falkowe o maksymalnej gładkości falki (falki Daubechies), to dodatkowo wykorzystujemy równania zerowania się momentów rzędu  $k$ .

$$\sum_{q=0}^{K-1} q^k (-1)^q h_{N-1-q} = 0, k = 0, 1, 2, \dots, K/2 - 1 \quad (\text{TF-6})$$

Przykład PF: Przekształcenie falkowe Daubechies 4 (czwartego rzędu).

W tym przypadku warunki ortogonalności macierzy  $P$  przyjmują postać:

$$h_0^2 + h_1^2 + h_2^2 + h_3^2 = 1, h_0 h_2 + h_1 h_3 = 0 \quad (\text{TF-7})$$

Równania zerowania momentów zerowego i pierwszego rzędu przyjmują postać:

$$h_3 - h_2 + h_1 - h_0 = 0, 0h_3 - 1h_2 + 2h_1 - 3h_0 = 0 \quad (\text{TF-8})$$

Rozwiązanie powyższego układu równań daje elementy filtru  $H$ :

$$H = \left[ \frac{1 + \sqrt{3}}{4\sqrt{2}}, \frac{3 + \sqrt{3}}{4\sqrt{2}}, \frac{3 - \sqrt{3}}{4\sqrt{2}}, \frac{1 - \sqrt{3}}{4\sqrt{2}} \right], G = [h_3, -h_2, h_1, -h_0] \quad (\text{TF-9})$$

Filtr  $H$  dla falki rzędu szóstego i ósmego mają następujące wartości:

$$H_{DB6} = [0.47046721, 1.14111692, 0.650365, -0.19093442, -0.12083221, 0.0498175] \quad (\text{TF-10})$$

$$H_{DB8} = [0.32580343, 1.01094572, 0.8922014, -0.03957503, -0.26450717, 0.0436163, 0.0465036, -0.01498699] \quad (\text{TF-11})$$