

Zadanie na wykonanie Projektu Zespołowego

Celem projektu jest uzyskanie następującego szeregu umiejętności praktycznych:

- umiejętności opracowania równoległych wersji algorytmów (na przykładzie algorytmów algebry liniowej i cyfrowego przetwarzania sygnałów);
- umiejętności projektowania wyspecjalizowanych urządzeń równoległych i potokowych (akceleratorów) na poziomach strukturalnym i logicznym;
- umiejętności opracowania modułów składowych systemów komputerowego wspomaganie projektowania (CAD) w/w algorytmów i akceleratorów;
- umiejętności pracy w zespole projektantów.

Ze względu na dość szeroki obszar podejmowanych zagadnień projekt został podzielony na dwie bezpośrednio nie powiązanych ze sobą części, które można wykonywać niezależnie. Opis poszczególnych części jest przedstawiony niżej.

Zadanie do części 1.

W ramach tej części należy opracować projekt wyspecjalizowanego urządzenia potokowego realizującego algorytm szybkiego przetwarzania (transformację) Fouriera (*ang.* FFT – Fast Fourier Transformation) na wektorze danych wejściowych $X(N)$, w którym każdy element $x_i \in X(N)$ jest liczbą zespoloną, tj. $x_i = \text{Re}x_i + j \cdot \text{Im}x_i$, gdzie $i = 1, 2, \dots, N = 2^n$, $j^2 = -1$, a n - dowolna liczba naturalna. Urządzenie powinno zawierać maksimum dwa bloki pamięci typu RAM (dla przechowania danych wejściowych i wyników), jeden blok pamięci stałej ROM (dla przechowania zespolonych współczynników obrotu $W_i = \text{Re}W_i + j \cdot \text{Im}W_i$ algorytmu FFT, gdzie $i = 1, 2, \dots, N/2$), potokową jednostkę przetwarzającą (ALU) o zadanej liczbie bloków mnożenia i sumatorów oraz blok sterowania wraz z układami dodatkowymi (np. multiplekserami, rejestrami, itd.) pozwalającymi na prowadzenie obliczeń w trybie potokowym. Urządzenie ma funkcjonować w oparciu o algorytm FFT o podstawie 2 z podziałem w dziedzinie częstotliwości lub czasu, z odwróconą bitowo kolejnością odczytywania danych wejściowych lub zapisywania wyników i wykonywać operację bazową FFT (motylek) w określonym czasie. Wszystkie w/w parametry urządzenia określone są (w zależności od numeru zadania) w oparciu o tab.1.

Proponuje się następujący plan pracy nad pierwszą częścią projektu.

Najpierw wykonuje się projekt potokowej jednostki przetwarzającej (ALU) realizującej operację bazową FFT i sporządza się tablica pracy ALU dla wybranej długości N wektora danych wejściowych, np. dla $N = 16$. Następnie formuje się schemat całego urządzenia, tj. do ALU zostają dodane bloki pamięci RAM, ROM oraz dodatkowe układy pozwalające na prowadzenie przez urządzenie obliczeń w trybie potokowym. To pozwala na formowanie specyfikacji (wymagań do) bloku sterowania urządzeniem. Następnie wykonuje się projekt bloku sterowania, włącznie z generatorami formującymi adresy zapisu/odczytu dla wszystkich bloków pamięci RAM i ROM. Następnie członkowie zespołu tworzą model bloku sterowania (np. w języku VHDL) i sprawdzają jego działania wykorzystując Active-VHDL, MatLab lub inne narzędzia programowe. Ze względu na obszerność zagadnień proponuje się wykonywać projekt w trzyosobowych zespołach.

Wiadomości teoretyczne na temat algorytmu FFT można znaleźć w ogólnie dostępnej literaturze, np.

1. R. G. Lyons. Wprowadzenie do cyfrowego przetwarzania sygnałów. WKŁ, 1999 r.
2. C. Marven, G. Ewers. Zarys cyfrowego przetwarzania sygnałów. WKŁ, 1999 r.
3. T. H. Cormen, C. E. Leiserson, R.L.Rivest, C. Stein. Wprowadzenie do algorytmów. WNT, 2005 r.

oraz w Internecie. W związku z tym, w niniejszym pliku umieszczono tylko informacje niezbędne do realizacji projektu, m. in. przykładowe grafy algorytmów 16-punktowego ($N=16$) FFT o podstawie 2 z podziałem w dziedzinie częstotliwości i czasu, z odwróconej bitowo kolejnością

odczytywania lub zapisywania danych, wzory opisujące operację bazowe FFT, tablicę tab.2 ilustrującą sposób formowania odwróconej bitowo kolejności adresów dla układów pamięci, przykładowy schemat ALU, fragment bloku sterowania oraz przykładowy schemat całego urządzenia.

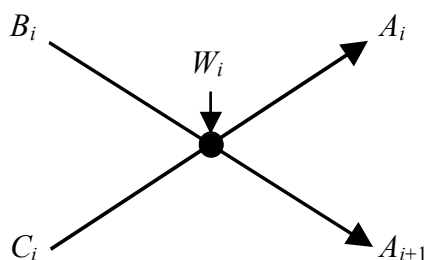
Tab.1 Tablica wariantów zadania dla części 1 projektu

Nr zad.	FFT z podziałem w dziedzinie: częstotliwości (F), czasu (T)	Liczba bloków mnożenia i sumatorów χ, Σ	Odwrócona bitowo kolejność danych: na wejściu (We), na wyjściu (Wy)	Maksymalna długość cyklu obliczeniowego (taktów zegara t)
1	F	1, 1	We	6
2	F	1, 2	We	4
3	F	2, 2	We	3
4	F	2, 3	We	2
5	F	2, 4	We	2
6	T	2, 4	We	2
7	F	1, 1	Wy	6
8	F	1, 2	Wy	4
9	F	2, 2	Wy	3
10	F	2, 3	Wy	2
11	F	2, 4	Wy	2
12	T	2, 4	Wy	2
13	F	2, 6	Wy	2
14	F	2, 6	We	2
15	T	1, 2	Wy	4
16	T	1, 2	We	4

I.

FFT o podstawie 2 z podziałem w czasie, (FFT radix-2, DIT)

Operacja bazowa (motylek)



$$A_i = B_i + C_i \cdot W_i$$

$$A_{i+1} = B_i - C_i \cdot W_i$$

B_i oraz C_i – wartości wejściowe,
 A_i oraz A_{i+1} – przeobrażone wartości,
 W_i – współczynnik obrotu (waga)

Wyrażenia, które muszą być obliczone w ALU (operacja bazowa)

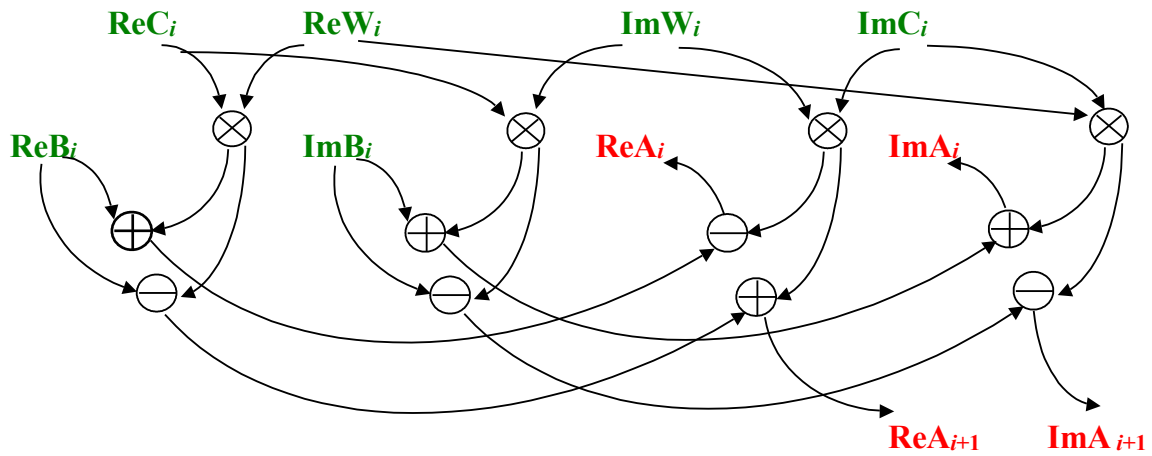
$$\text{Re } A_i = \text{Re } B_i + \text{Re } C_i \cdot \text{Re } W_i - \text{Im } C_i \cdot \text{Im } W_i$$

$$\text{Im } A_i = \text{Im } B_i + \text{Re } C_i \cdot \text{Im } W_i + \text{Im } C_i \cdot \text{Re } W_i$$

$$\text{Re } A_{i+1} = \text{Re } B_i - \text{Re } C_i \cdot \text{Re } W_i + \text{Im } C_i \cdot \text{Im } W_i$$

$$\text{Im } A_{i+1} = \text{Im } B_i - \text{Re } C_i \cdot \text{Im } W_i - \text{Im } C_i \cdot \text{Re } W_i$$

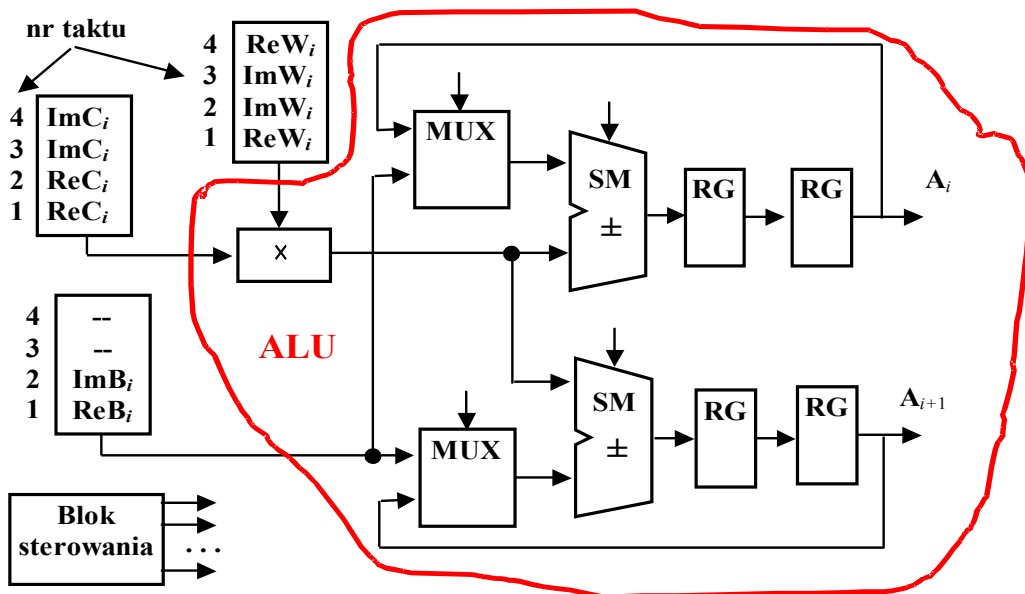
Graf operacji bazowej



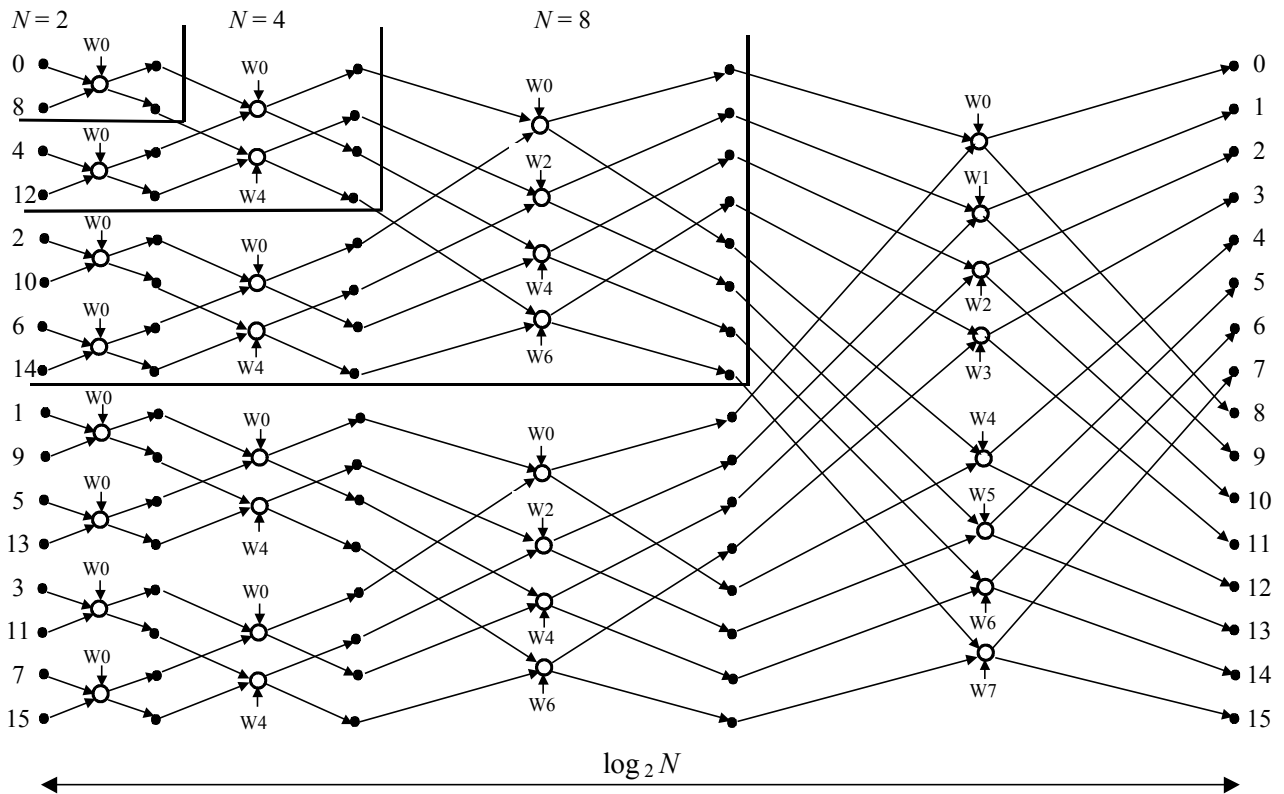
Schemat potokowego ALU realizującego operację bazową (zaznaczono czerwonym kolorem)

Parametry podstawowe:

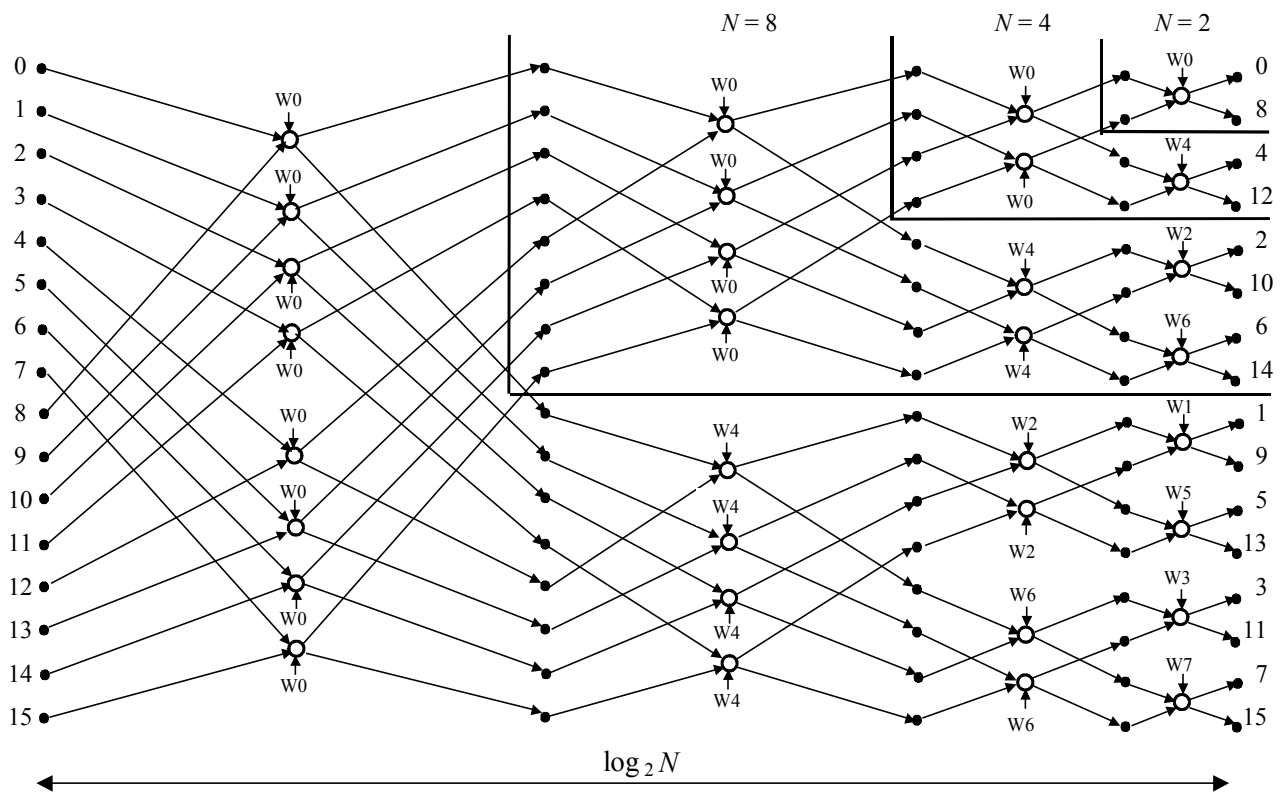
- 1 blok mnożący;
- 2 sumatory;
- czas wykonania operacji bazowej (długość cyklu obliczeniowego) – 4 takty zegarowe



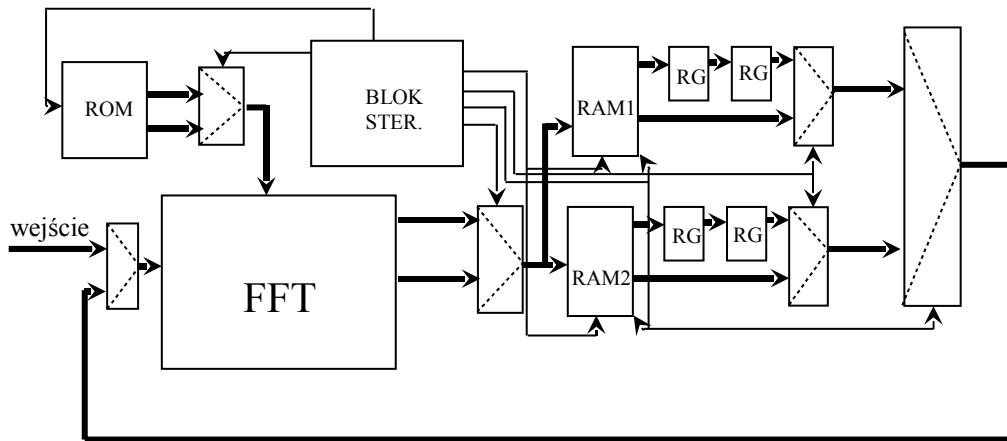
Graf 16-punktowego FFT o podstawie 2, z podziałem w czasie
odwrócona bitowo kolejność danych wejściowych i normalnie uporządkowane wyniki
 (16-point FFT; radix-2; DIT; bit reversed input data order; normally ordered output data)



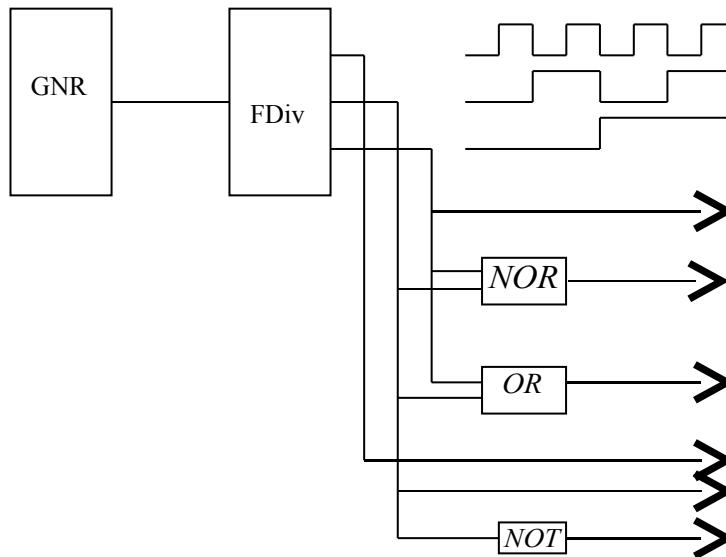
Graf 16-punktowego FFT o podstawie 2, z podziałem w czasie
odwrócona bitowo kolejność wyników i normalnie uporządkowane dane wejściowe
 (16-point FFT; radix-2; DIT; bit reversed output data order; normally ordered input data)



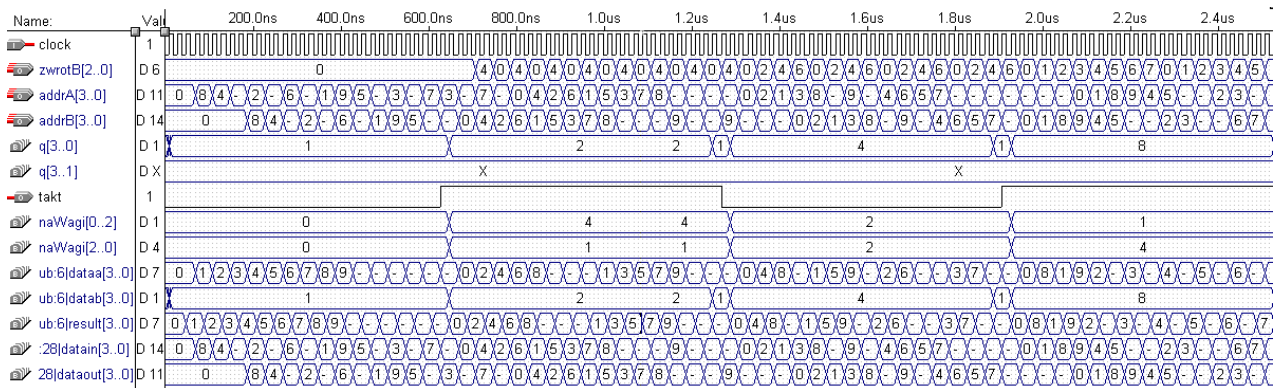
Schemat ogólny urządzenia



Fragment bloku sterowania



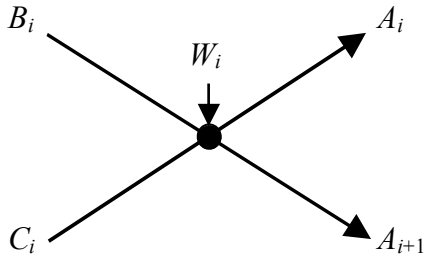
Wyniki symulacji działania bloku sterowania i generatorów adresu



II.

FFT o podstawie 2 z podziałem w dziedzinie częstotliwości
FFT radix-2, DIF

Operacja bazowa (motylek)



$$A_i = B_i + C_i$$

$$A_{i+1} = (B_i - C_i) \cdot W_i$$

B_i oraz C_i – wartości wejściowe,
 A_i oraz A_{i+1} – przetworzone wartości,
 W_i – współczynnik obrotu (waga)

Wyrażenia, które muszą być obliczone w ALU (operacja bazowa):

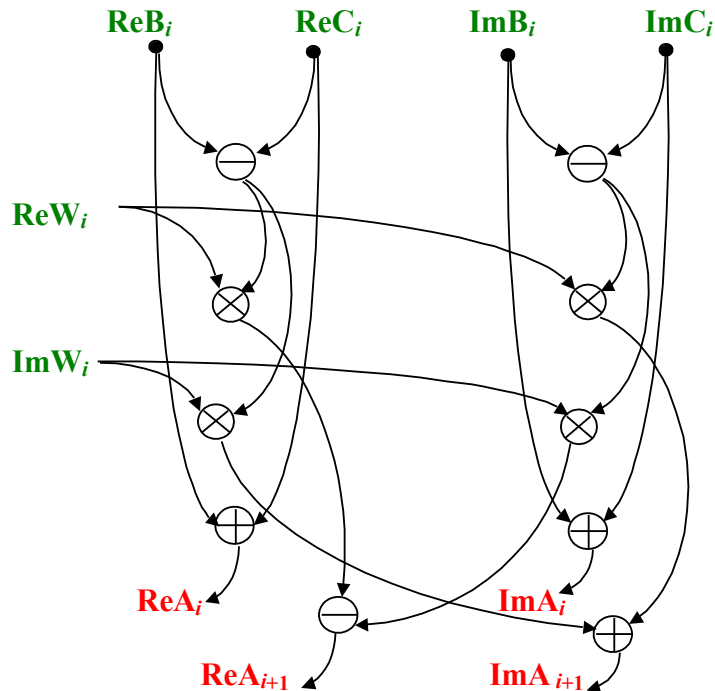
$$\text{Re } A_i = \text{Re } B_i + \text{Re } C_i$$

$$\text{Im } A_i = \text{Im } B_i + \text{Im } C_i$$

$$\text{Re } A_{i+1} = (\text{Re } B_i - \text{Re } C_i) \cdot \text{Re } W_i - (\text{Im } B_i - \text{Im } C_i) \cdot \text{Im } W_i$$

$$\text{Im } A_{i+1} = (\text{Re } B_i - \text{Re } C_i) \cdot \text{Im } W_i + (\text{Im } B_i - \text{Im } C_i) \cdot \text{Re } W_i$$

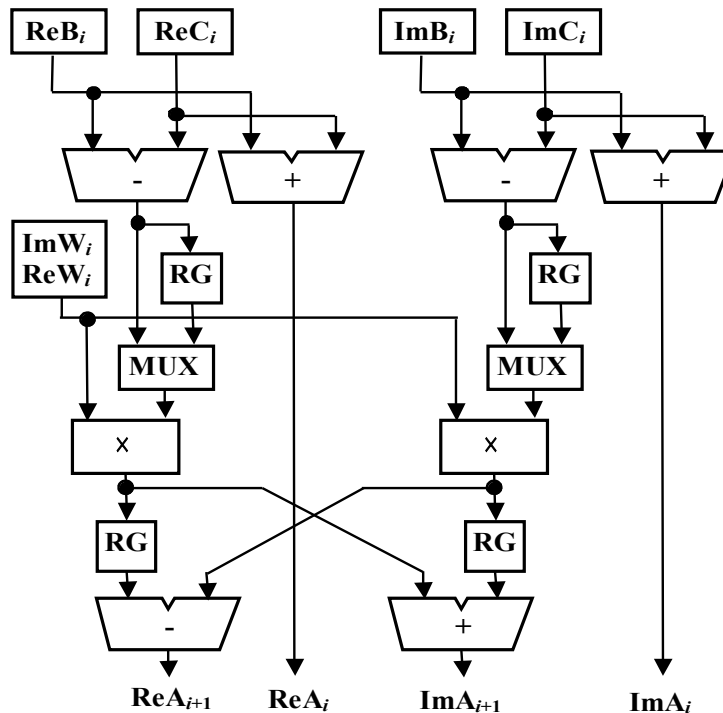
Graf operacji bazowej



Schemat potokowego ALU realizującego operację bazową

Parametry podstawowe:

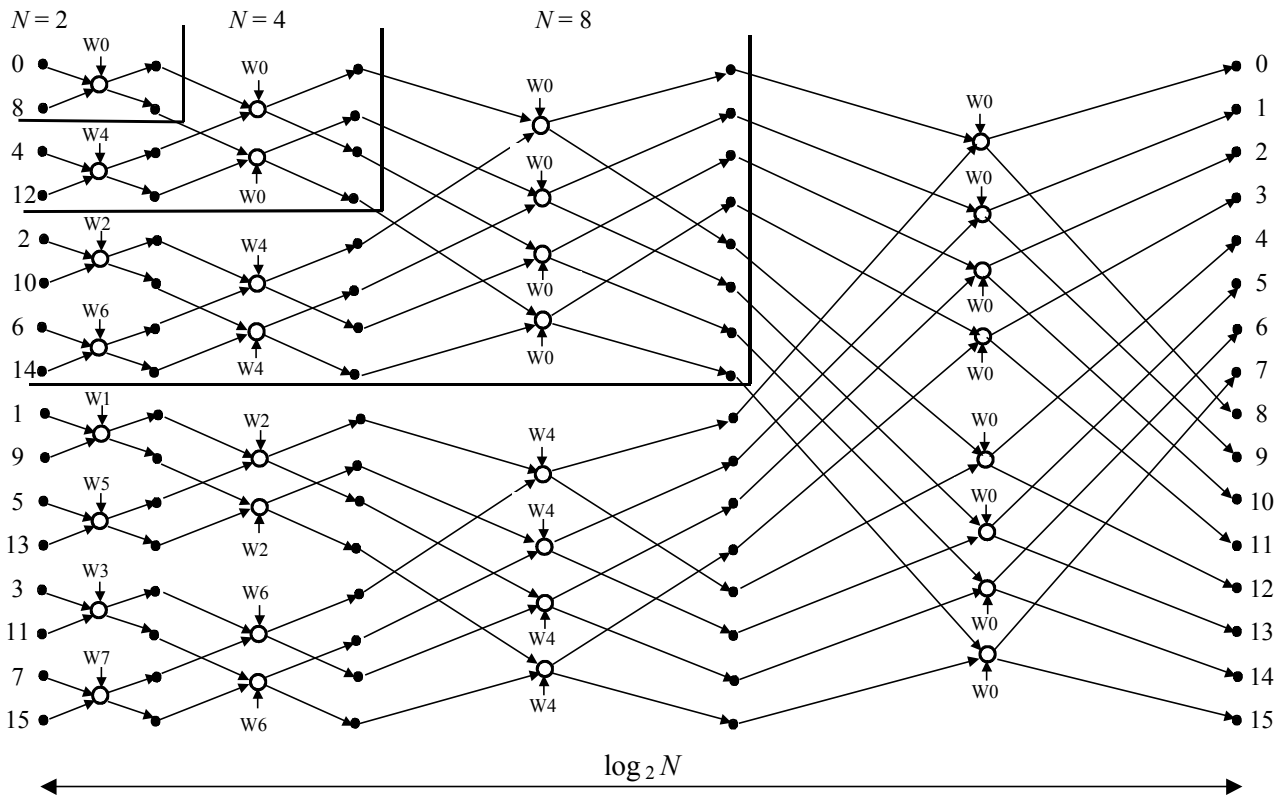
- 2 bloki mnożące;
- 6 sumatorów;
- czas wykonania operacji bazowej (długość cyklu obliczeniowego) – 2 takty zegarowe



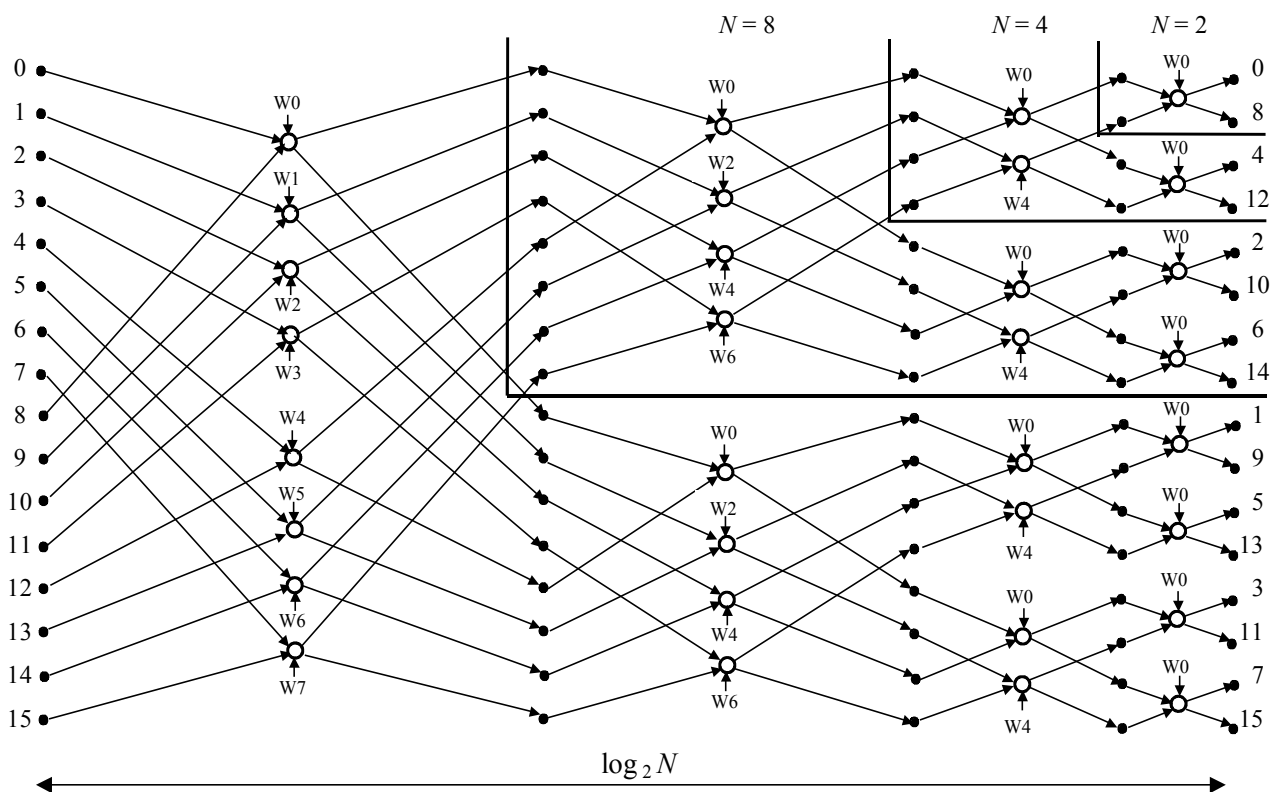
Tab. 2. Zasada formowania odwróconej bitowo kolejności adresów danych We/Wy

Nr kroku (liczba kroków wynosi $\log_2 N = 4$)							
1	2	3	4	5	6	7	8
0000	0	0000	0	0000	0	0000	0
1000	8	0100	4	0010	2	0001	1
0100	4	0010	2	0001	1	1000	8
1100	12	0110	6	0011	3	1001	9
0010	2	0001	1	1000	8	0100	4
1010	10	0101	5	1010	10	0101	5
0110	6	0011	3	1001	9	1100	12
1110	14	0111	7	1011	11	1101	13
0001	1	1000	8	0100	4	0010	2
1001	9	1100	12	0110	6	0011	3
0101	5	1010	10	0101	5	1010	10
1101	13	1110	14	0111	7	10110	11
0011	3	1011	9	1100	12	0110	6
1011	11	1101	13	1110	14	0111	7
0111	7	1011	11	1101	13	1110	14
1111	15	1111	15	1111	15	1111	15

Graf 16-punktowego FFT o podstawie 2, z podziałem w dziedzinie częstotliwości odwrócona bitowo kolejność danych wejściowych i normalnie uporządkowane wyniki (16-point FFT; radix-2; DIF; bit reversed input data order; normally ordered output data)



Graf 16-punktowego FFT o podstawie 2, z podziałem w dziedzinie częstotliwości odwrócona bitowo kolejność wyników i normalnie uporządkowane dane wejściowe (16-point FFT; radix-2; DIF; bit reversed output data order; normally ordered input data)



Zadanie do części 2.

W ramach tej części projektu należy opracować dwa podstawowe moduły środowiska komputerowego wspomagania projektowania równoległych wersji algorytmów regularnych oraz architektur równoległych akceleratorów dla ich realizacji. Algorytm wejściowy należy do grupy algorytmów algebry liniowej i jest zadany przy pomocy fragmentu programu zawierającego jedno lub kilka gniazd pętli o różnej złożoności (tj. zawierającego różną liczbę włożonych instrukcji pętli).

Co należy zrobić:

1. Zgodnie z otrzymanym od prowadzącego numerem wariantu zadania wybrać z tab. 3 odpowiedni algorytm *AL*. Należy zapoznać się z jego opisem i grafem (informacje te umieszczono na końcu tego pliku).
2. Zapoznać się z metodą konstruowania grafów zależności informacyjnych algorytmów zadanych za pomocą włożonych instrukcji pętli, zwracając szczególną uwagę na sposób uzyskania współrzędnych dla poszczególnych jego wierzchołków oraz na sposoby otrzymania listy jego łuków (w/w metoda będzie szczegółowo omawiana na wykładzie z przedmiotu „Projektowanie systemów informatycznych”).
3. Opracować moduł *GRAF* wchodzący do składu środowiska CAD, którego zadaniem jest generowanie opisu grafu algorytmu *AL* (listę wierzchołków i łuków) dla różnych rozmiarów N macierzy danych wejściowych, np. $N = 3 \div 10$. Zespół może dodatkowo opracować i umieścić w programie *GRAF* moduł umożliwiający wizualizację grafu *AL*.
4. Opracować moduł *DESIGNER* (lub ewentualnie rozbudować moduł *GRAF*) realizujący metodę odwzorowania n - wymiarowego grafu algorytmu w $(n-1)$ - wymiarowe architektury akceleratorów równoległych (gdzie n – największy wymiar gniazda pętli w algorytmie). Wyżej wymieniona metoda będzie szczegółowo omawiana na wykładzie z przedmiotu „Projektowanie systemów informatycznych”.
5. Korzystając z zaprojektowanych modułów *GRAF* i *DESIGNER* zaprojektować dwie $(n - 1)$ - wymiarowe architektury akceleratorów równoległych, z których pierwsza ma największy współczynnik obciążenia elementów przetwarzających EP (lub największe przyspieszenie), a druga jest lepsza od pierwszej pod względem innego (jednego lub kilku) kryterium np.:
 - * liczba EP;
 - * czas wykonania algorytmu;
 - * liczba kanałów We/Wy (zewnętrznych i wewnętrznych);
 - * liczba różnych typów EP.

Opracować programy wykonawcze lub tablice pracy dla jednego dowolnego procesora w każdej z zaprojektowanych architektur.

Tabela 3.

<i>AL</i>	Nazwa metody lub zagadnienia	Uwagi
1	Rozkład LU macierzy metodą Gaussa	Kolejność obliczeń : według wierszy macierzy
2	Rozkład LU macierzy metodą Gaussa	Kolejność obliczeń : według kolumn macierzy
3	Rozkład LL^T macierzy metodą Cholesky'ego	Symetryczna macierz wejściowa
4	Eliminacja Gaussa $M \cdot A = A^*$	Kolejność obliczeń : według wierszy macierzy
5	Eliminacja Gaussa $M \cdot A = A^*$	Kolejność obliczeń : według kolumn macierzy
6	Rozkład QR macierzy metodą Givensa $Q \cdot A = R$	Macierz prostokątna
7	Rozkład QR macierzy metodą Givensa $Q \cdot A = R$	Macierz kwadratowa Hessenberga
8	Rozkład QR macierzy metodą Givensa $Q \cdot A = R$	Macierz kwadratowa pasmowa Hessenberga
9	Redukcja wsteczna (rozwiązywanie układu równań liniowych $A \cdot x = b$)	Macierz górna trójkątna pasmowa. Kolejność obliczeń: według wierszy macierzy
10	Metoda podstawienia (rozwiązywanie układu równań liniowych $A \cdot x = b$)	Macierz dolna trójkątna pasmowa. Kolejność obliczeń: według kolumn macierzy
11	Rozwiązanie układu równań $A \cdot X = B$ metodą Jordana-Gaussa	Kilka wektorów wyrazów wolnych. Kolejność obliczeń: według wierszy macierzy
12	Rozwiązanie układu równań $A \cdot X = B$ metodą Jordana-Gaussa	Kilka wektorów wyrazów wolnych. Kolejność obliczeń: według kolumn macierzy

Krótki opis algorytmów

1. Rozkład LU macierzy metodą Gaussa

Dane wejściowe: macierz $A(N,N)$

Wyniki: $L(N,N)$ - dolna macierz trójkątna, $U(N,N)$ - górna macierz trójkątna, takie, że $A = L \cdot U$.

$$\boxed{A(N,N)} = \boxed{\begin{array}{c} \diagup \\ L \quad U \end{array}}$$

for $i:=1$ to $N-1$ **do**

begin

for $j:=i+1$ to N **do**

if $a_{ii} \neq 0$ **then** $l_{ji} := a_{ji} / a_{ii}$

else $l_{ji} := 0$;

for $j:=i+1$ to N **do**

for $k:=i+1$ to N **do**

$a_{jk} := a_{jk} - l_{ji} \cdot a_{ik}$;

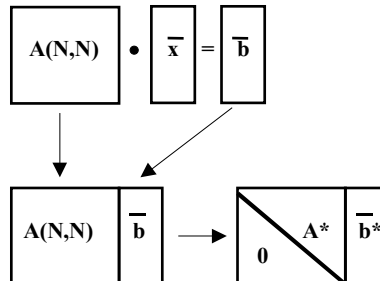
end

Wyniki : $l_{ii} = 1$; $u_{ij} = a_{ij}$, dla $i \leq j$.

2. Metoda eliminacji Gaussa przekształcenia macierzy kwadratowej do postaci macierzy górnej trójkątnej.

Dane wejściowe: macierz $\mathbf{A}(N,N)$ i wektor wyrazów wolnych $\mathbf{b}(N)$ równania $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$, które razem formują macierz rozszerzoną $\mathbf{A}^*(N, N+1)$ (patrz rysunek).

Wyniki: $\mathbf{A}^*(N,N)$ - górna macierz trójkątna, $\mathbf{b}^*(N)$ - wektor, takie, że $\mathbf{M} \cdot \mathbf{A} = \mathbf{A}^*$, $\mathbf{M} \cdot \mathbf{b} = \mathbf{b}^*$ (gdzie \mathbf{M} - dolna macierz trójkątna)



```

for i:=1 to N-1 do
  begin
    for j:=i+1 to N do
      if  $a_{ii} \neq 0$  then  $m_{ji} := -a_{ji} / a_{ii}$ 
      else  $m_{ji} := 0$ ;
    for j:=i+1 to N do
      for k:=i+1 to N+1 do
         $a_{jk} := a_{jk} + m_{ji} \cdot a_{ik}$  ;
      end
  end

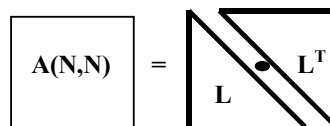
```

Wyniki: $a^*_{ij} = a_{ij}$, dla $i \leq j$;
 $b_i^* = a_{i,N+1}$.

3. Rozkład LL^T macierzy metodą Cholesky'ego.

Dane wejściowe: macierz $\mathbf{A}(N,N)$ symetryczna.

Wyniki: dolna trójkątna macierz $\mathbf{L}(N,N)$, taka że $\mathbf{A} = \mathbf{L} \cdot \mathbf{L}^T$



```

for i := 1 to N do
  begin
     $a_{ii} := \text{SQRT}(a_{ii})$ ;
    for j := i+1 to N do
       $a_{ji} := a_{ji} / a_{ii}$ ;
    for j := i+1 to N do
      for k := i+1 to j do
         $a_{jk} := a_{jk} - a_{ji} * a_{ki}$ ;
      end;
  end;

```

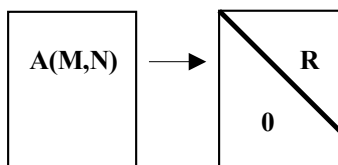
Wyniki: $l_{ij} := a_{ij}$ dla $i \geq j$.

4. Rozkład QR macierzy metodą Givensa

Dane wejściowe: macierz $\mathbf{A}(M,N)$ ($N \leq M$)

Wyniki: macierz górna trójkątna \mathbf{R}

```
for i:=1 to N do
  begin
    for j:=i+1 to M do
      begin
         $a_{ii} := \text{sqrt}((a_{ii})^2 + (a_{ji})^2);$ 
         $c_{ji} := a_{ii} / a_{ii};$ 
         $s_{ji} := a_{ji} / a_{ii};$ 
      end;
    for j:=i+1 to M do
      for k:=i+1 to N do
        begin
           $\text{temp} := c_{ji} \cdot a_{ik} + s_{ji} \cdot a_{jk};$ 
           $a_{jk} := -s_{ji} \cdot a_{ik} + c_{ji} \cdot a_{jk};$ 
           $a_{ik} := \text{temp};$ 
        end
      end
    end
end
```

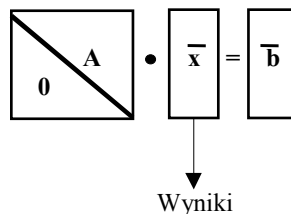


Wyniki: $r_{ij} = a_{ij}$ dla $i \leq j$

5. Redukcja wsteczna (rozwiązanie układu równań $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$ z trójkątną macierzą)

Dane wejściowe: macierz $\mathbf{A}(N,N)$ górna trójkątna, wektor $\mathbf{b}(N)$.

Wyniki: wektor niewiadomych $\mathbf{x}(N)$

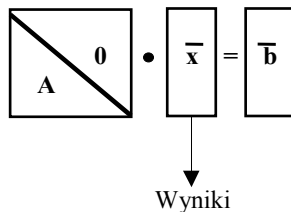


```
for i := N downto 1 do
  begin
     $x_i := b_i / a_{ii};$ 
    for j := i-1 downto 1 do
       $b_j := b_j - a_{ji} \cdot x_i;$ 
    end;
  end;
```

6. Metoda podstawienia (rozwiązanie układu równań $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$ z macierzą trójkątną)

Dane wejściowe: macierz $\mathbf{A}(N,N)$ dolna trójkątna, wektor $\mathbf{b}(N)$.

Wyniki: wektor niewiadomych $\mathbf{x}(N)$



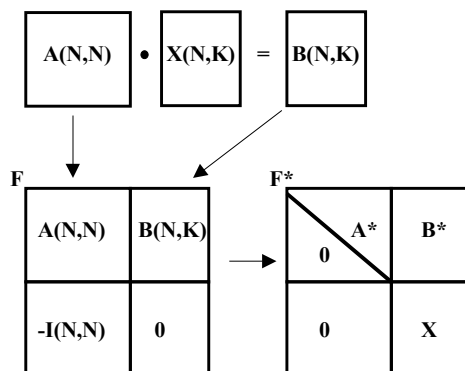
```

for  $i := 1$  to  $N$  do
  begin
     $x_i := b_i / a_{ii}$ ;
    for  $j := i+1$  to  $N$  do
       $b_j := b_j - a_{ji} \cdot x_i$ ;
    end;
  
```

7. Rozwiązanie układu równań lub odwracania macierzy metodą Jordana-Gaussa $\mathbf{A} \cdot \mathbf{X} = \mathbf{B}$.

Dane wejściowe: macierz współczynników $\mathbf{A}(N,N)$, macierz wyrazów wolnych $\mathbf{B}(N,K)$ które wraz z macierzami jednostkową \mathbf{I} i zerową \mathbf{O} tworzą macierz \mathbf{F} (patrz rysunek).

Wyniki: macierz $\mathbf{X}(N,K)$



```

for  $i := 1$  to  $N$  do
  begin
    for  $j := i+1$  to  $N+i$  do
       $m_{ji} := -f_{ji} / f_{ii}$ ;
    for  $j = i+1$  to  $N+i$  do
      for  $k = i+1$  to  $N+K$  do
         $f_{jk} = f_{jk} + m_{ji} \cdot f_{ik}$ ;
    end;
  
```

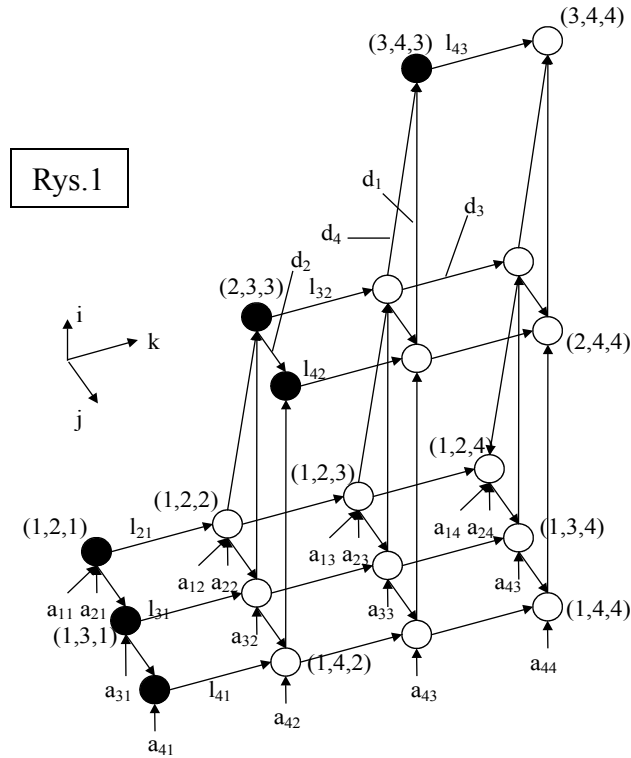
gdzie $f_{ij} = a_{ij}$, $i=1,2,\dots,N$, $j=1,2,\dots,N$;
 $f_{ij} = b_{ij}$, $i=1,2,\dots,N$, $j=N+1,N+2,\dots,N+K$;
 $f_{(N+i)i} = -1$, $i=1,2,\dots,N$;
 $f_{(N+i)j} = 0$, $i=1,2,\dots,N$, $j=i+1,i+2,\dots,N+K$;

Wyniki: $x_{ij} = f_{(N+i)(N+j)}$, $i=1,2,\dots,N$, $j=1,2,\dots,K$.

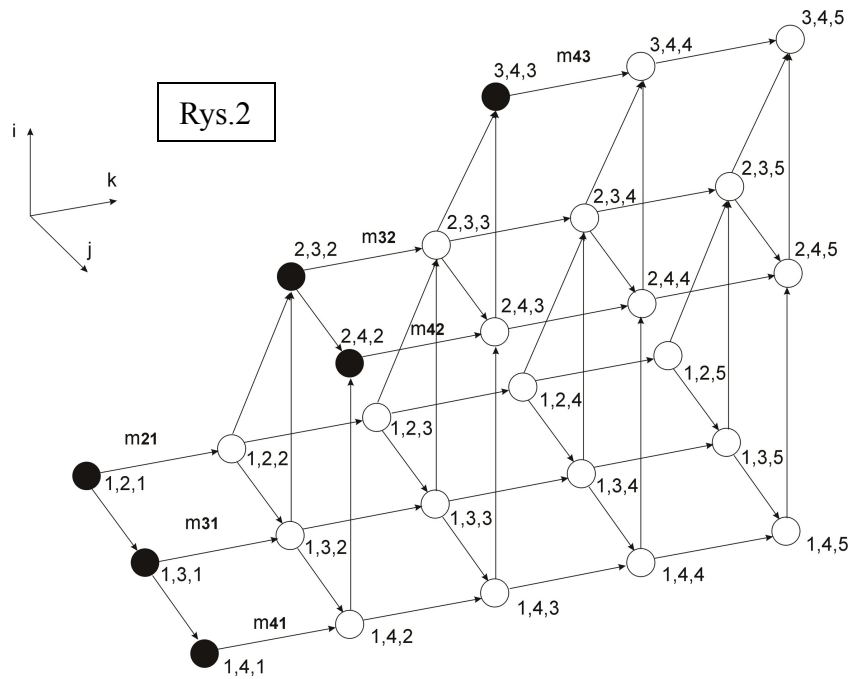
Grafy wybranych algorytmów

(opracowane przez studentów)

Rozkład LU macierzy $A(4,4)$ metodą Gaussa

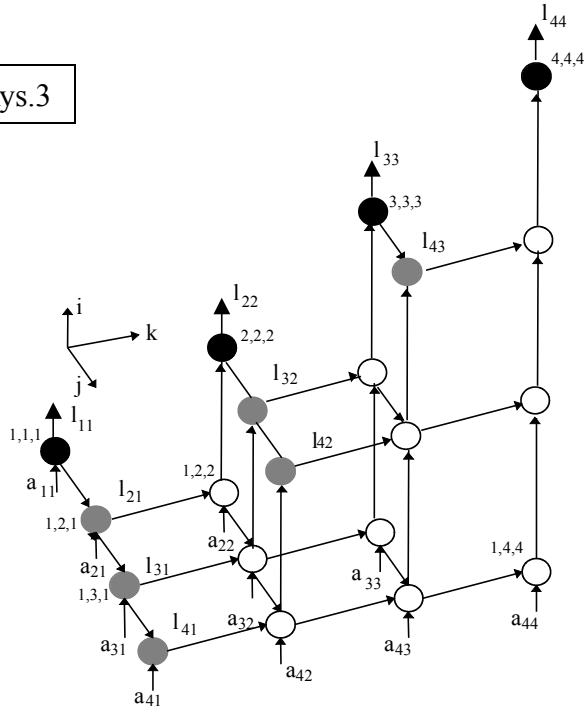


Eliminacja Gaussa $M \cdot A(4,4) = A^*$, $M \cdot b(4) = b^*$



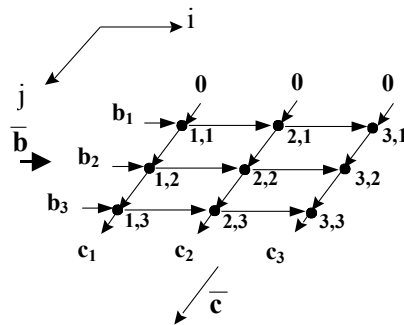
LL^T - dekompozycja macierzy $A(4,4)$ metodą Cholesky'ego

Rys.3



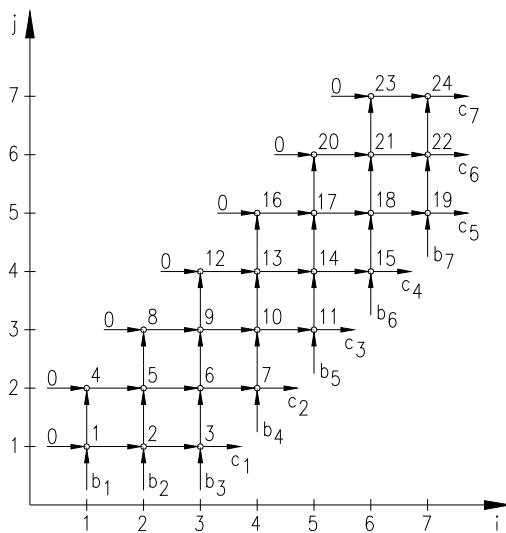
Mnożenie macierzy przez wektor $A(3,3) \cdot b(3) = c(3)$

Rys.4



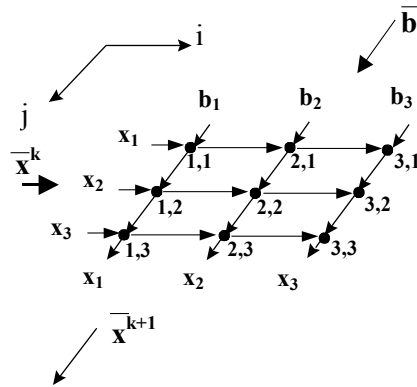
Przypadek mnożenia macierzy kwadratowej pasmowej $A(7,7) \cdot b(7) = c(7)$, szerokość pasma $L=4$

Rys.5



Rozwiązanie układu równań metodą iteracji prostej $x' = D(3,3) \cdot x(3) + b(3)$

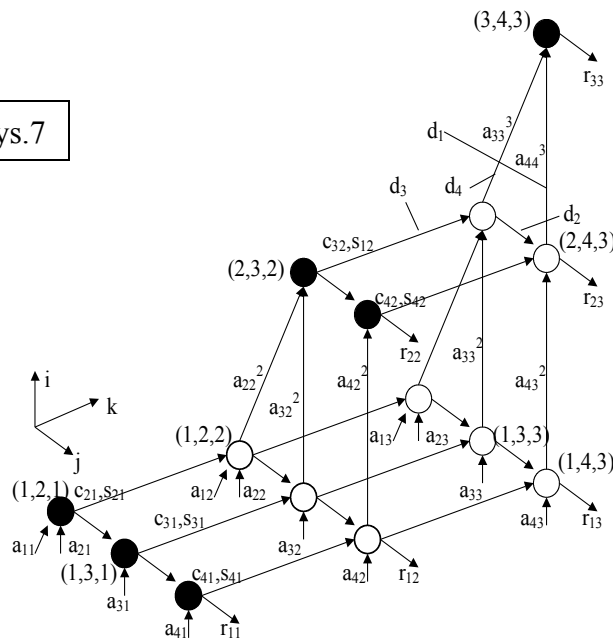
Rys.6



W przypadku macierzy kwadratowej pasmowej graf będzie podobny do grafu przedstawionego na Rys.5 (mnożenie macierzy pasmowej przez wektor)

QR – dekompozycja macierzy metodą Givensa $Q \cdot A = R$ (Macierz prostokątna $A(4,3)$)

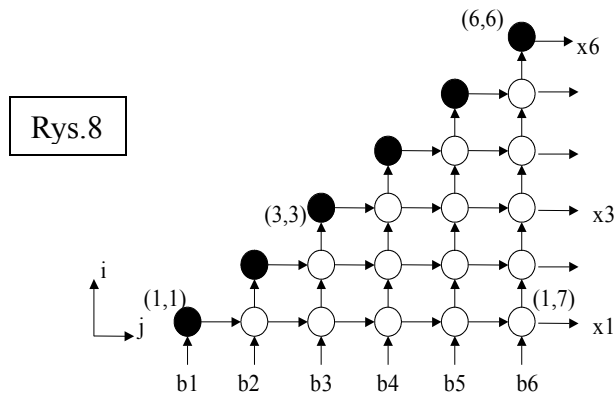
Rys.7



(W przypadku macierzy kwadratowej Hessenberga graf będzie podobny do grafu przedstawionego na Rys.8)

Metoda podstawienia dla rozwiązywania układu równań liniowych $A(6,6) \cdot x(6) = b(6)$

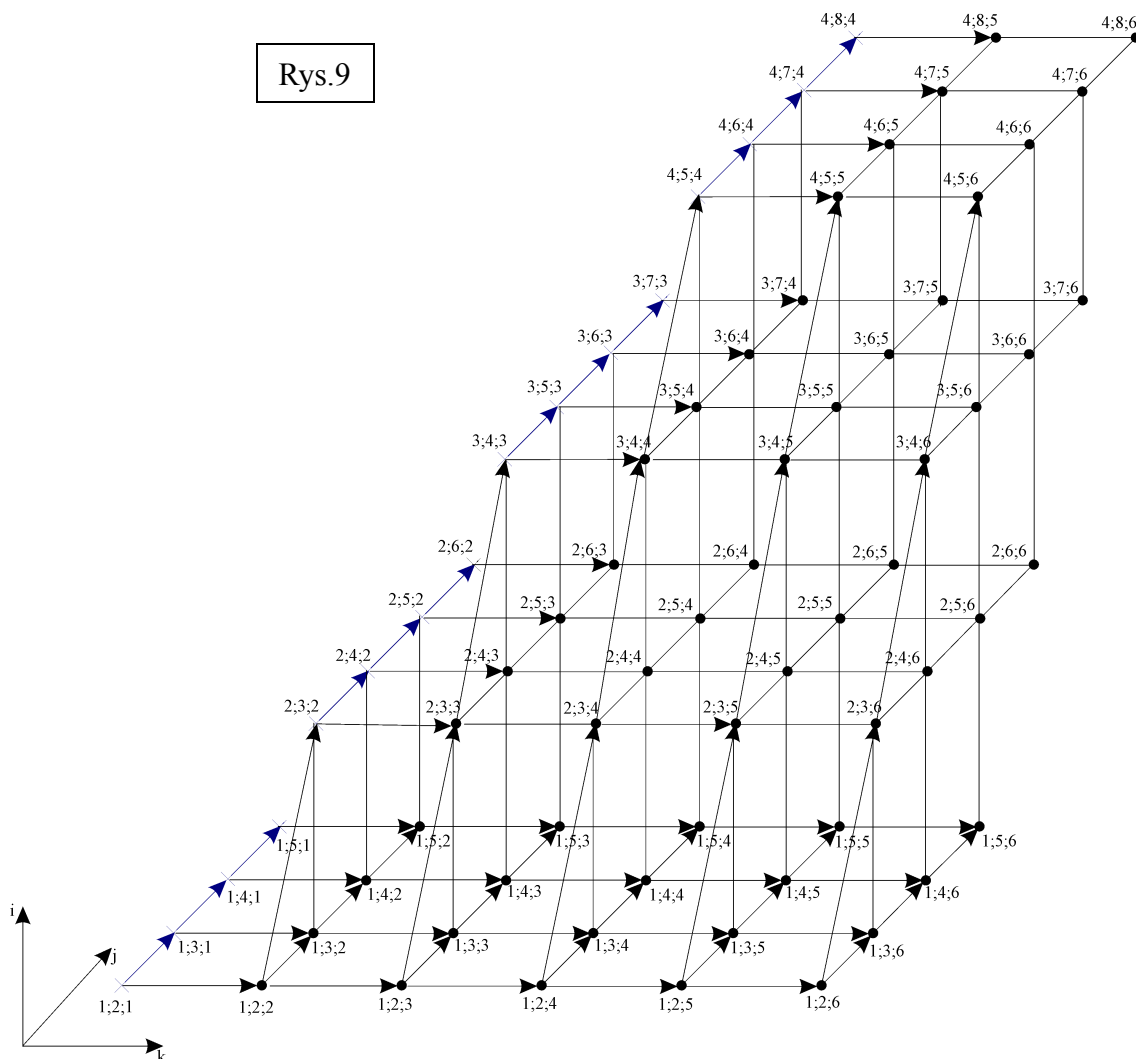
(Dla macierzy pasmowej: patrz mnożenie macierzy pasmowej przez wektor)



Redukcja wsteczna rozwiązywania układu równań liniowych $A \cdot x = b$

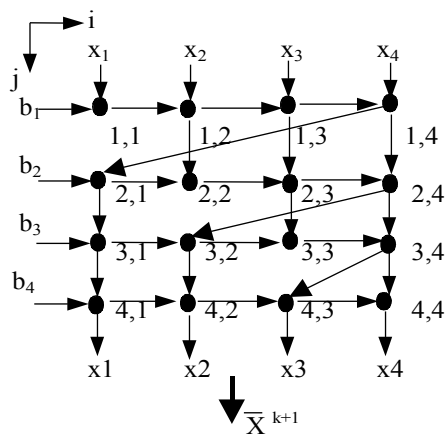
(Graf jest podobny do grafu przedstawionego na Rys.8)

Rozwiązanie układu równań metodą Jordana-Gaussa $A(4,4) \cdot X(4,2) = B(4,2)$



Rozwiązanie układu równań metodą Gaussa-Seidela $A(4,4) \cdot x(4) = b(4)$

Rys.10



Splot dwóch funkcji (filtracja jednowymiarowa)

Rys.11

