

Systemy przetwarzania sygnałów

Zrównoleglanie procedur CPS



wykład No 11

Podstawowym wymogiem zadań CPS jest **czas przetwarzania**, czyli możliwość dokonania przewidzianej symulacji w dopuszczalnym dla danego zastosowania czasie.

Główną cechą systemów CPS jest możliwość ich pracy w trybie czasu rzeczywistego, czyli konieczność wykonania wszystkich operacji obliczeniowych i sterujących w czasie wystarczającym na niezakłócony przebieg trwającego procesu.

Sposoby minimalizacji czasu realizacji zadań CPS



Przetwarzanie potokowe (ang. pipelining)

Przetwarzanie potokowe jest jednym ze sposobów równoległego przetwarzania danych informatycznych.

- Sposób przetwarzania danych polegający na wykonaniu po kolei serii programów, których wyniki przekazywane są do innego programu lub urządzenia.

- Sposób pracy procesora, który polega na wykonywaniu kolejnych instrukcji bez czekania na zakończenie poprzednich, jeżeli te same zasoby procesora nie są potrzebne do ich wykonania.

Sposób działania: cykl przetwarzania dzieli się na odrębne bloki przetwarzania, z których każdy oprócz pierwszego i ostatniego jest połączony z następnym. Dane po przejściu przez jeden blok trafiają do następnego, aż osiągną ostatni blok.

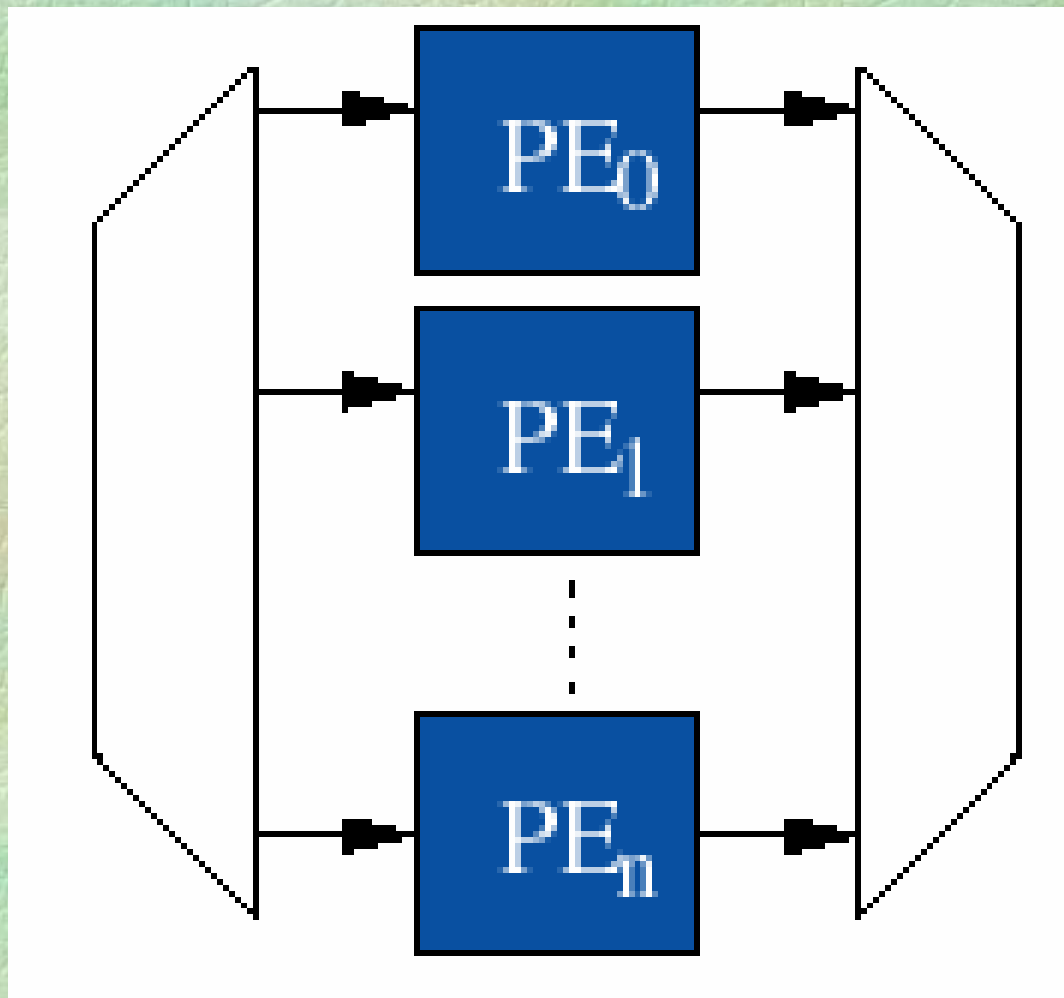


Dzięki temu, że przetwarzanie odbywa się w rozdzielnych blokach, system może przetwarzać jednocześnie tyle danych ile zdefiniowano bloków.

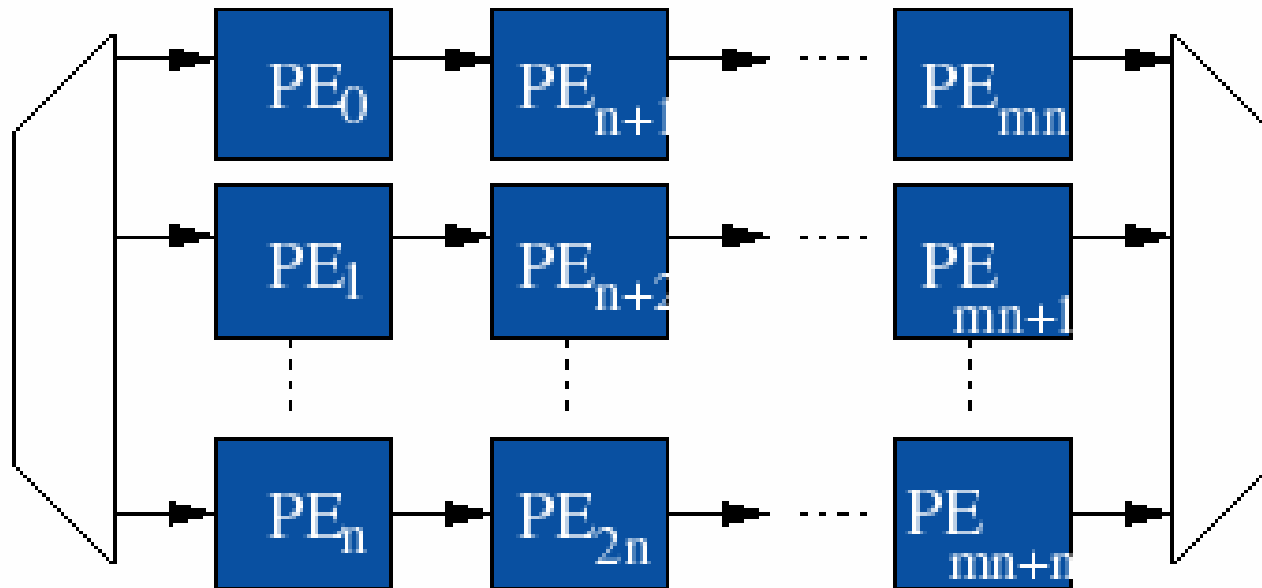
Przetwarzanie współbieżne (ang. *Parallel Processing*), zwane także równoległym, umożliwia wykonywanie większej ilości zadań w tym samym czasie, poprzez zwiększenie ilości procesorów w systemie komputerowym.

Przetwarzanie równoległe - jest to wykonywanie kilku czynności (zadań) w tym samym czasie przez różne procesory. Jest to węższe pojęcie od przetwarzania współbieżnego.

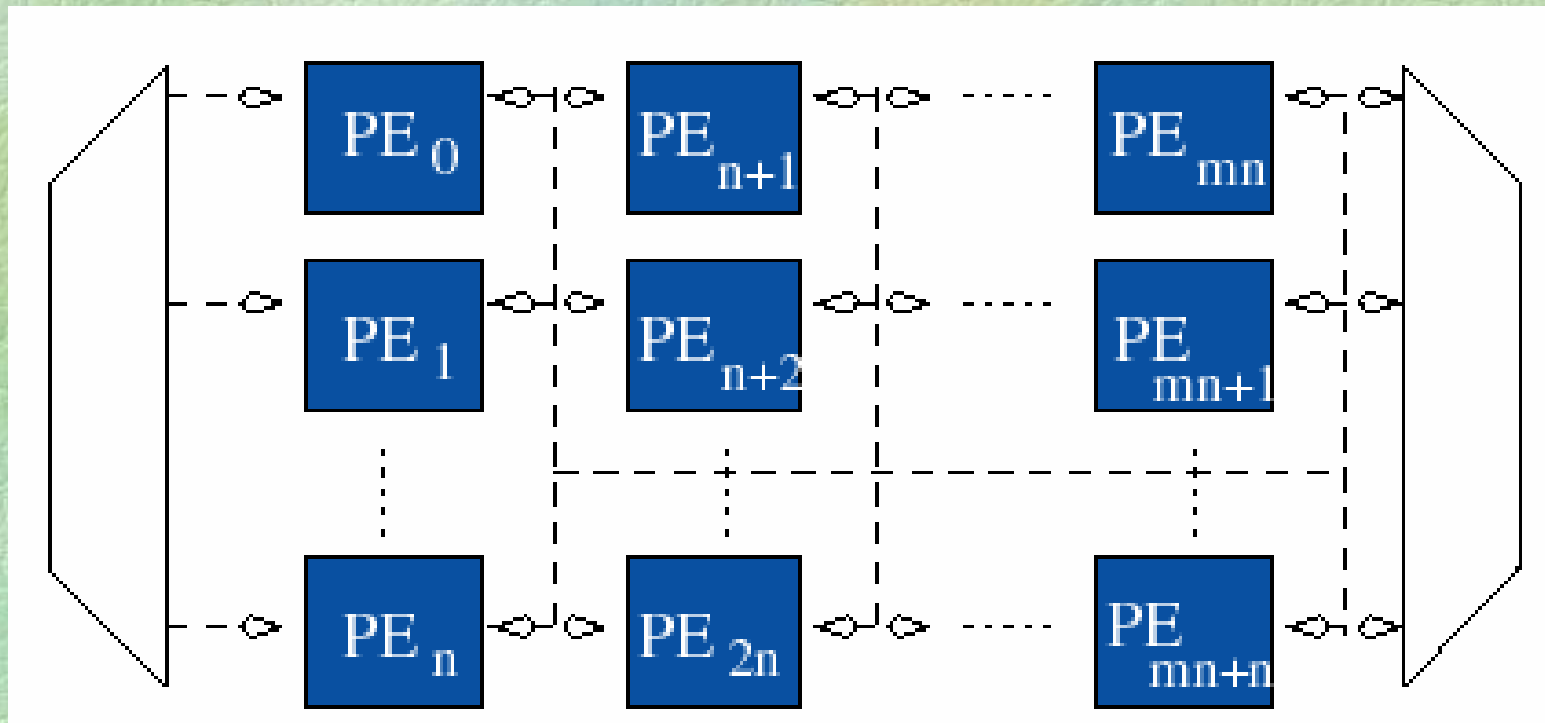
Równoległy

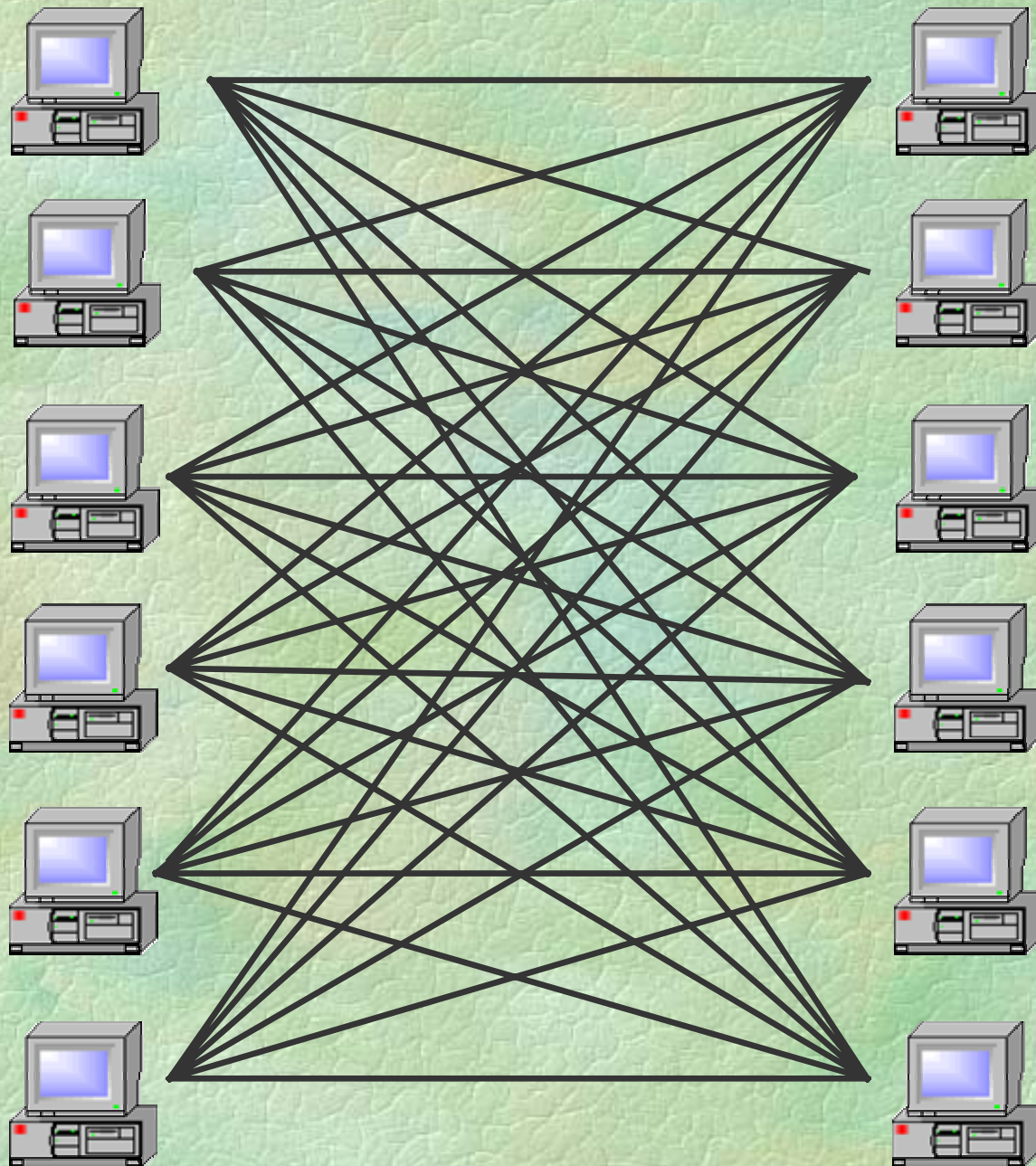


Wielopotokowy



Hybrydowy





Równoległe algorytmy obliczania iloczynu wektorowo-macierzowego

- bloki powstają w wyniku podziału macierzy pionowymi oraz poziomymi liniami w taki sposób aby liczba wierszy i kolumn każdego bloku była identyczna,

- bloki składają się z pewnej jednakowej liczby wierszy i powstają poprzez podział macierzy liniami poziomymi, liczba kolumn w bloku jest identyczna jak dzielonej macierzy,

- bloki składają się z pewnej jednakowej liczby kolumn i powstają poprzez podział macierzy liniami pionowymi, liczba wierszy w bloku jest identyczna jak dzielonej macierzy.

$$\mathbf{Z}_{N \times 1} = [z_0, z_1, \dots, z_{N-1}]^T$$

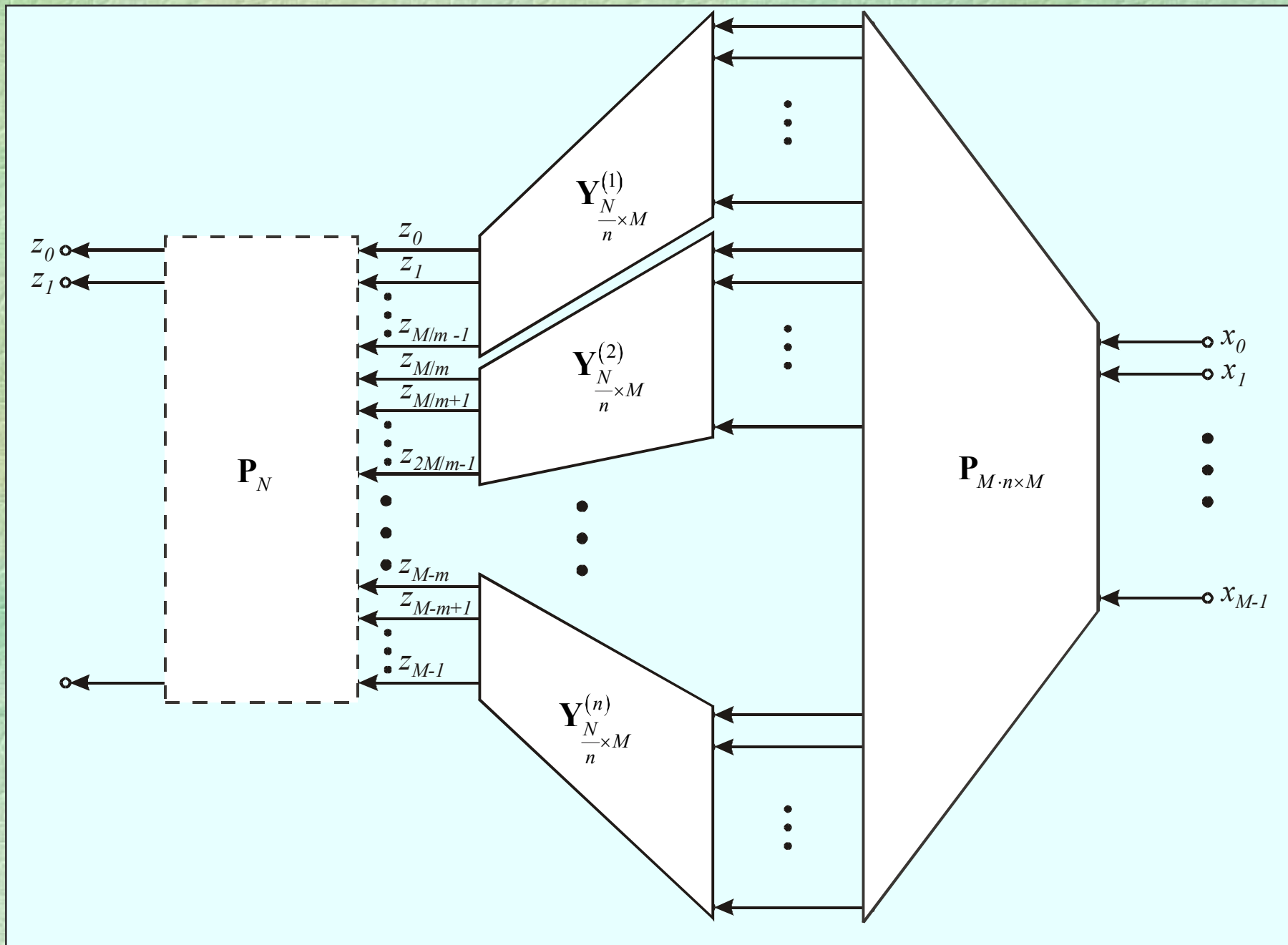
$$\mathbf{Y}_{N \times M} = \|y_{i,j}\|$$

$$\mathbf{X}_{M \times 1} = [x_0, x_1, \dots, x_{M-1}]^T$$

$$\mathbf{Z}_{N \times 1} = \begin{bmatrix} \vdots \\ \mathbf{Y}_{\frac{N}{n} \times M}^{(i)} \\ \vdots \end{bmatrix} \times \mathbf{X}_{M \times 1}$$

$$\mathbf{Y}_{N \times M} = \begin{bmatrix} \mathbf{Y}_{\frac{N}{n} \times M}^{(1)} \\ \text{---} & \text{---} & \text{---} \\ \mathbf{Y}_{\frac{N}{n} \times M}^{(2)} \\ \text{---} & \text{---} & \text{---} \\ \vdots \\ \mathbf{Y}_{\frac{N}{n} \times M}^{(n)} \end{bmatrix}$$

$$\mathbf{Y}_{\frac{N}{n} \times M}^{(i)} = \begin{bmatrix} y_{(i-1) \cdot \frac{N}{n}, 0} & y_{(i-1) \cdot \frac{N}{n}, 1} & \cdots & y_{(i-1) \cdot \frac{N}{n}, M-1} \\ y_{(i-1) \cdot \frac{N}{n} + 1, 0} & y_{(i-1) \cdot \frac{N}{n} + 1, 1} & \cdots & y_{(i-1) \cdot \frac{N}{n} + 1, M-1} \\ \vdots & \vdots & \ddots & \vdots \\ y_{i \cdot \frac{N}{n}, 0} & y_{i \cdot \frac{N}{n}, 1} & \cdots & y_{i \cdot \frac{N}{n}, M-1} \end{bmatrix}$$



$$\mathbf{Z}_{N \times 1} = [z_0, z_1, \dots, z_{N-1}]^T$$

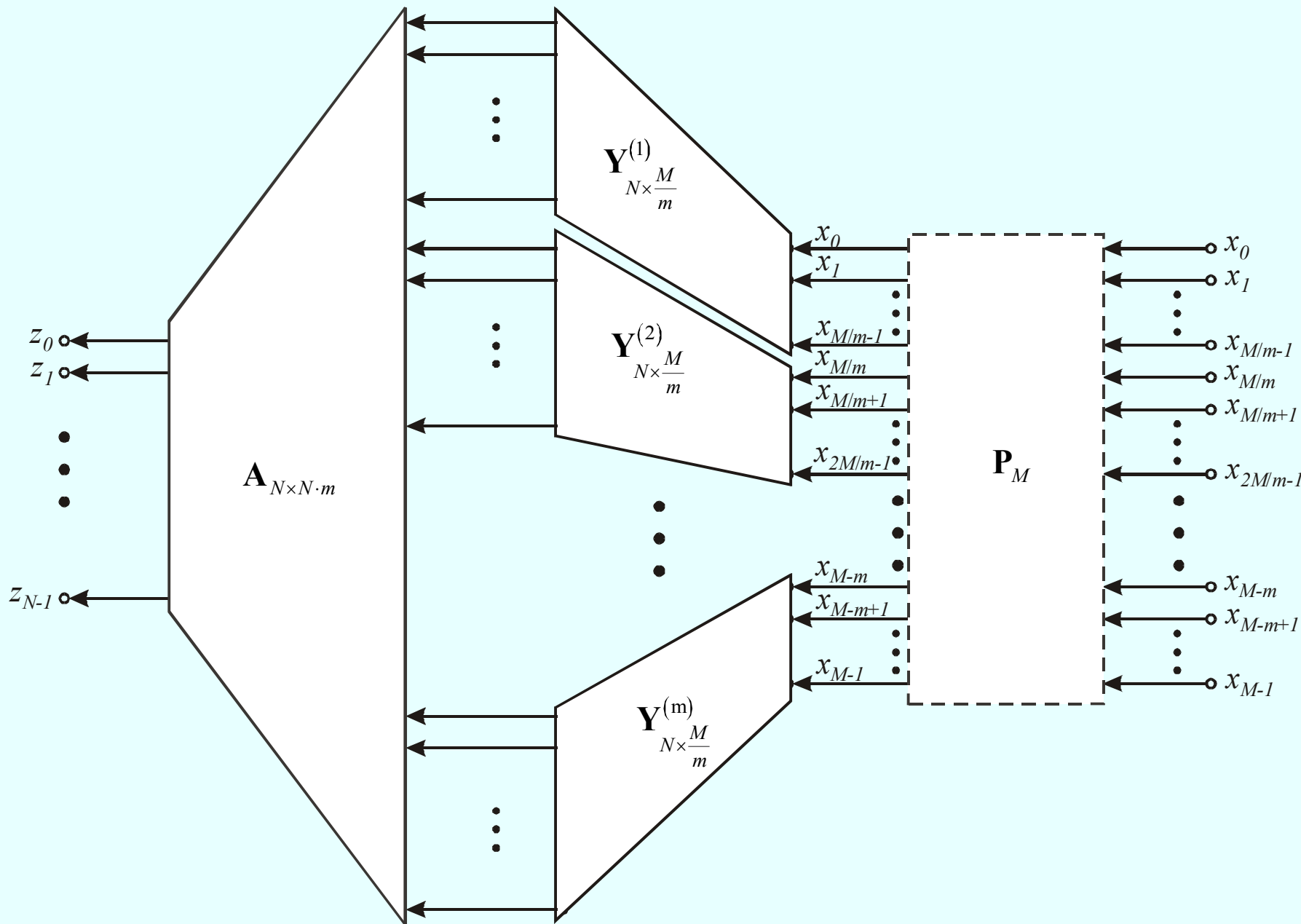
$$\mathbf{Y}_{N \times M} = \|y_{i,j}\|$$

$$\mathbf{X}_{M \times 1} = [x_0, x_1, \dots, x_{M-1}]^T$$

$$\mathbf{Z}_{N \times 1} = \begin{bmatrix} \vdots & \dots & \mathbf{Y}_{N \times \frac{M}{m}}^{(j)} & \dots & \vdots \end{bmatrix} \times \mathbf{X}_{M \times 1}$$

$$\mathbf{Y}_{N \times M} = \left[\begin{array}{c|c|c|c} \mathbf{Y}_{N \times \frac{M}{m}}^{(1)} & \mathbf{Y}_{N \times \frac{M}{m}}^{(2)} & \cdots & \mathbf{Y}_{N \times \frac{M}{m}}^{(m)} \end{array} \right]$$

$$\mathbf{Y}_{N \times \frac{M}{m}}^{(j)} = \left[\begin{array}{cccc} y_{0,(j-1) \cdot \frac{M}{m}} & y_{0,(j-1) \cdot \frac{M}{m} + 1} & \cdots & y_{0,j \cdot \frac{M}{m} - 1} \\ y_{1,(j-1) \cdot \frac{M}{m}} & y_{1,(j-1) \cdot \frac{M}{m} + 1} & \cdots & y_{1,(j-1) \cdot \frac{M}{m} - 1} \\ \vdots & \vdots & \ddots & \vdots \\ y_{N-1,(j-1) \cdot \frac{M}{m}} & y_{N-1,(j-1) \cdot \frac{M}{m} + 1} & \cdots & y_{M-1,(j-1) \cdot \frac{M}{m} - 1} \end{array} \right]$$



$$\mathbf{Z}_{N \times 1} = [z_0, z_1, \dots, z_{N-1}]^T$$

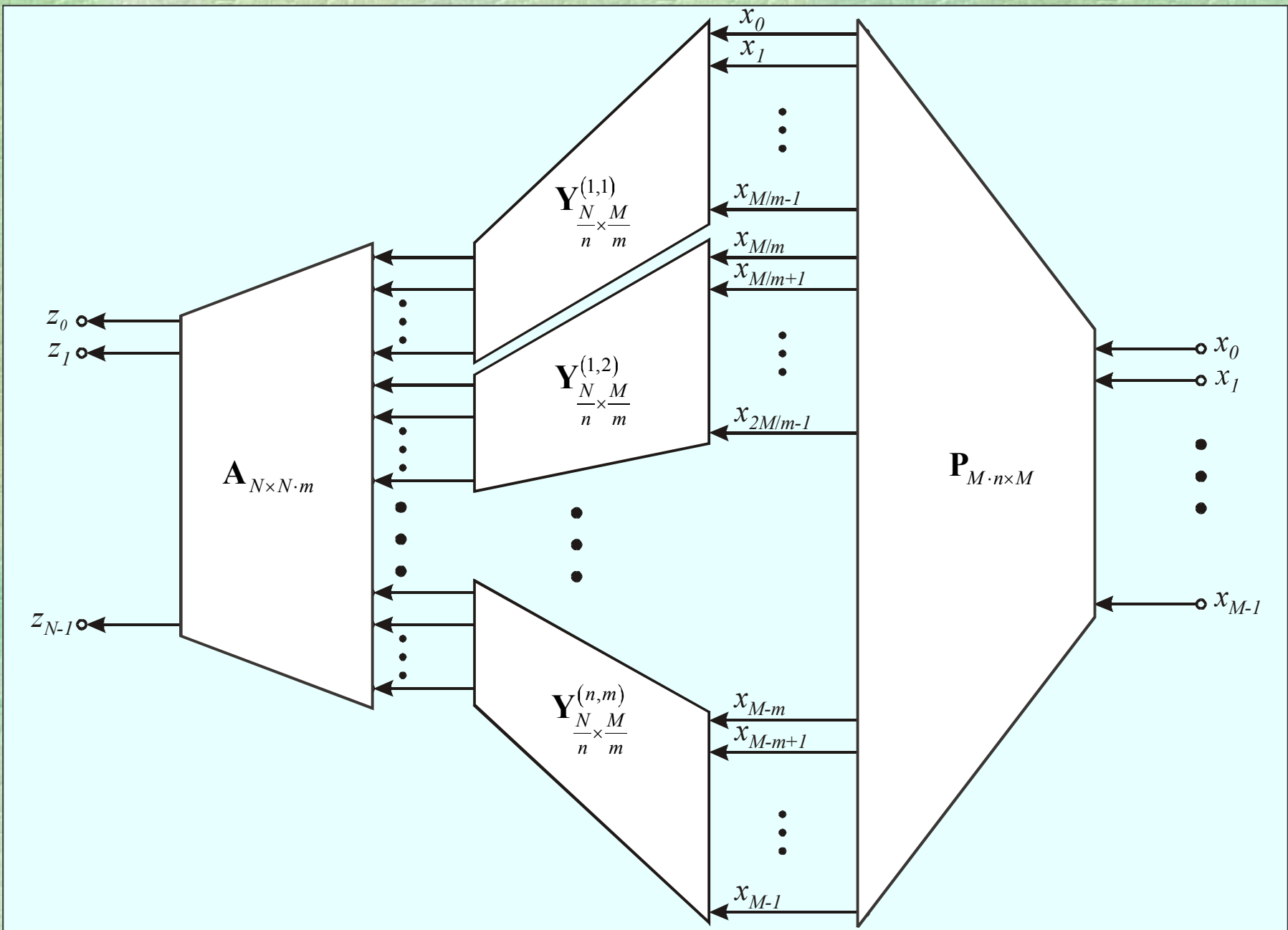
$$\mathbf{Y}_{N \times M} = \|y_{i,j}\|$$

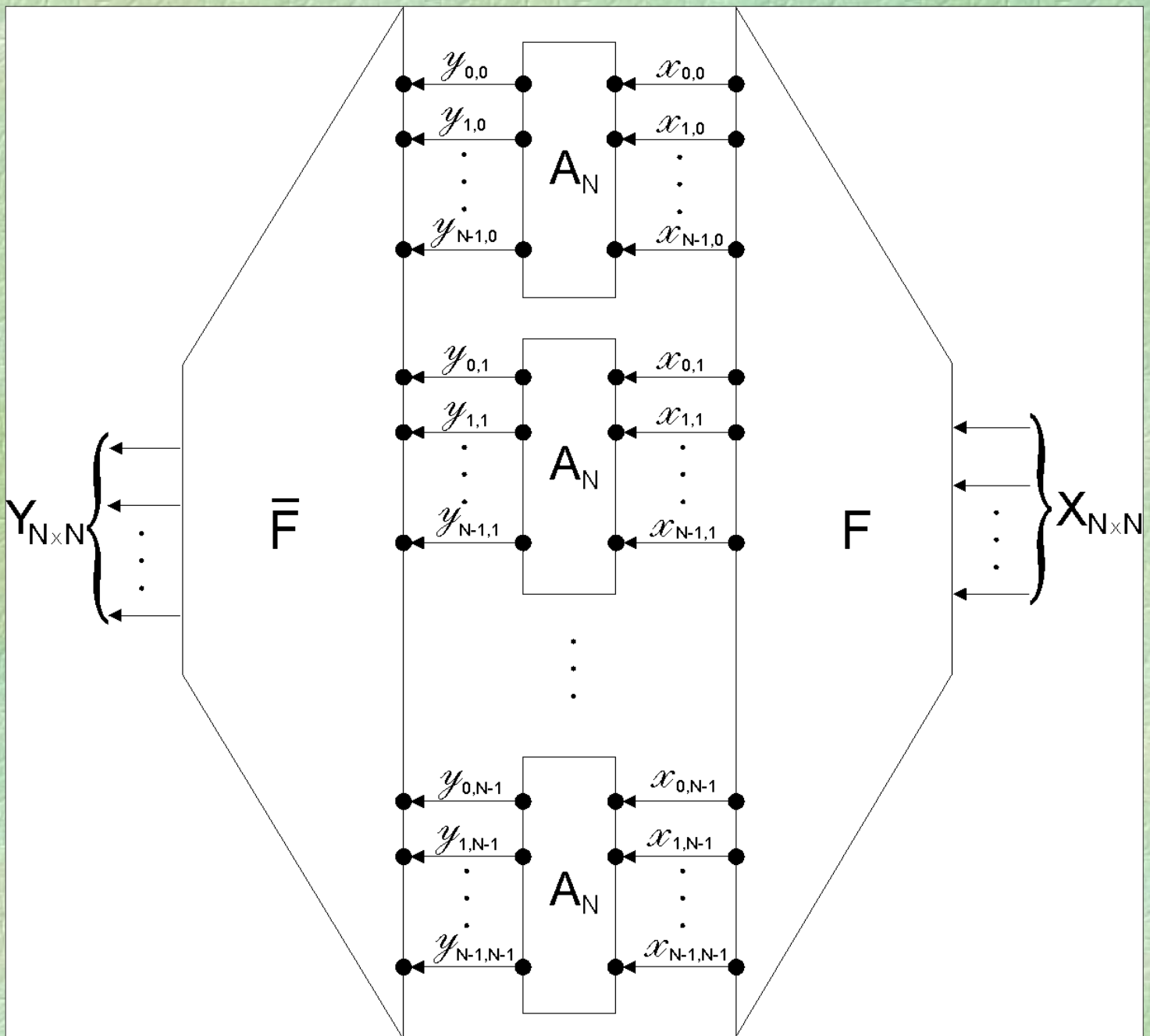
$$\mathbf{X}_{M \times 1} = [x_0, x_1, \dots, x_{M-1}]^T$$

$$\mathbf{Z}_{N \times 1} = \begin{matrix} \begin{array}{|c|} \hline \mathbf{Z}_{N \times 1} \\ \hline \end{array} \end{matrix} = \begin{matrix} \begin{array}{|c|c|c|c|c|} \hline & & & & \\ \hline & \begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} & \begin{array}{c} \vdots \end{array} & \begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} & \\ \hline & \begin{array}{c} \cdot \quad \cdot \quad \cdot \end{array} & \mathbf{Y}_{\frac{N}{n} \times \frac{M}{m}}^{(i,j)} & \begin{array}{c} \cdot \quad \cdot \quad \cdot \end{array} & \\ \hline & \begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} & \begin{array}{c} \vdots \end{array} & \begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} & \\ \hline & & & & \\ \hline \end{array} \end{matrix} \times \begin{matrix} \begin{array}{|c|} \hline \mathbf{X}_{M \times 1} \\ \hline \end{array} \end{matrix}$$

$$\mathbf{Y}_{N \times M} = \begin{bmatrix} \mathbf{Y}_{\frac{N}{n} \times \frac{M}{m}}^{(1,1)} & \mathbf{Y}_{\frac{N}{n} \times \frac{M}{m}}^{(1,2)} & \dots & \mathbf{Y}_{\frac{N}{n} \times \frac{M}{m}}^{(1,m)} \\ \mathbf{Y}_{\frac{N}{n} \times \frac{M}{m}}^{(2,1)} & \mathbf{Y}_{\frac{N}{n} \times \frac{M}{m}}^{(2,2)} & & \mathbf{Y}_{\frac{N}{n} \times \frac{M}{m}}^{(2,m)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{Y}_{\frac{N}{n} \times \frac{M}{m}}^{(n,1)} & \mathbf{Y}_{\frac{N}{n} \times \frac{M}{m}}^{(n,2)} & \dots & \mathbf{Y}_{\frac{N}{n} \times \frac{M}{m}}^{(n,m)} \end{bmatrix}$$

$$\mathbf{Y}_{\frac{N}{n} \times \frac{M}{m}}^{(i,j)} = \begin{bmatrix} y_{(i-1) \cdot \frac{N}{n}, (j-1) \cdot \frac{M}{m}} & y_{(i-1) \cdot \frac{N}{n}, (j-1) \cdot \frac{M}{m} + 1} & \dots & y_{(i-1) \cdot \frac{N}{n}, j \cdot \frac{M}{m}} \\ y_{(i-1) \cdot \frac{N}{n} + 1, (j-1) \cdot \frac{M}{m}} & y_{(i-1) \cdot \frac{N}{n} + 1, (j-1) \cdot \frac{M}{m} + 1} & \dots & y_{(i-1) \cdot \frac{N}{n} + 1, j \cdot \frac{M}{m}} \\ \vdots & \vdots & \ddots & \vdots \\ y_{i \cdot \frac{N}{n}, (j-1) \cdot \frac{M}{m}} & y_{i \cdot \frac{N}{n}, (j-1) \cdot \frac{M}{m} + 1} & \dots & y_{i \cdot \frac{N}{n}, j \cdot \frac{M}{m}} \end{bmatrix}$$





Równoległe algorytmy obliczenia splotu

$$\mathbf{X} = [x_0, x_1, \dots, x_{N-1}]^T$$

$$\mathbf{H} = [h_0, h_1, \dots, h_{M-1}]^T$$

$$\mathbf{Y} = [y_0, y_1, \dots, y_{N+M-2}]^T$$

$$\mathbf{Y}^{(j)} = \mathbf{A}_{(N+M-1) \times NM}^{(j)} \cdot \mathbf{H}_{NM}^{(j)} \cdot \mathbf{P}_{NM \times N}^{(j)} \cdot \mathbf{X}$$

$\mathbf{P}_{NM \times N}^{(j)}$ - macierz superwektoryzacji danych wejściowych

$\mathbf{H}_{NM}^{(j)}$ - macierz „współczynników wagowych” filtracji

$\mathbf{A}_{(N+M-1) \times NM}^{(j)}$ - macierz sumowania iloczynów

$$\mathbf{P}_{NM \times N}^{(1)} = \bigoplus_{\alpha=0}^{N-1} \mathbf{1}_M^{(\alpha)}$$

$$\mathbf{H}_{NM}^{(1)} = \mathbf{I}_N \otimes \mathbf{H}_M$$

$$\mathbf{H}_M = \text{diag}(h_0, h_1, \dots, h_{M-1})$$

$$\mathbf{A}_{(N+M-1) \times NM}^{(1)} = \begin{bmatrix} \blacksquare & \blacksquare \\ \vdots & \vdots \\ \blacksquare & \blacksquare \end{bmatrix}_{i=0}^{N-1} \mathbf{A}_{(N+M-1) \times M}^{(i)}$$

$$\mathbf{A}_{(N+M-1) \times M}^{(i)} = \mathbf{I}_{N+M-1}^{(i \rightarrow)} \cdot \mathbf{V}_{(N+M-1) \times M}$$

$$\mathbf{V}_{(N+M-1) \times M} = \mathbf{I}_M \begin{bmatrix} \blacksquare \\ \vdots \\ \blacksquare \end{bmatrix} \mathbf{0}_{(N-1) \times M}$$

$$\mathbf{P}_{NM \times M}^{(2)} = \left(\begin{array}{c} \blacksquare \\ \blacksquare \end{array} \begin{array}{c} M-1 \\ m=1 \end{array} \mathbf{V}_{m \times N}^{(m)} \right) \begin{array}{c} \blacksquare \\ \blacksquare \end{array} \begin{array}{c} N-M \\ \alpha=0 \end{array} \left(\begin{array}{c} \blacksquare \\ \blacksquare \end{array} \mathbf{W}_{M \times N}^{(\alpha)} \right) \left(\begin{array}{c} \blacksquare \\ \blacksquare \end{array} \begin{array}{c} M-1 \\ m=1 \end{array} \Lambda_{(M-m) \times N} \right)$$

$$\mathbf{V}_{m \times N}^{(m)} = \mathbf{I}_m \begin{array}{c} \blacksquare \\ \blacksquare \end{array} \mathbf{0}_{m \times (N-m)}$$

$$\mathbf{W}_{M \times N}^{(\alpha)} = \mathbf{W}_{M \times N} \cdot \mathbf{I}_N^{(\alpha \rightarrow)}$$

$$\mathbf{W}_{M \times N} = \mathbf{I}_M \begin{array}{c} \blacksquare \\ \blacksquare \end{array} \mathbf{0}_{M \times (N-M)}$$

$$\Lambda_{(M-1) \times N}^{(m)} = \mathbf{0}_{(M-m) \times (N-m)} \begin{array}{c} \blacksquare \\ \blacksquare \end{array} \mathbf{I}_{M-m}$$

$$\mathbf{A}_{(N+M-1) \times NM}^{(2)} = \left(\bigoplus_{m=1}^{M-1} \mathbf{1}_m^T \right) \oplus (\mathbf{I}_{N-M+1} \otimes \mathbf{1}_M^T) \oplus \left(\bigoplus_{m=1}^{M-1} \mathbf{1}_{M-m}^T \right)$$

$$\mathbf{H}_{NM}^{(2)} = \left(\bigoplus_{m=1}^{M-1} \mathbf{D}^{(m)} \right) \oplus (\mathbf{I}_{N-M+1} \otimes \left(\bigoplus_{k=1}^M h_{M-k} \right)) \oplus \left(\bigoplus_{m=1}^{M-1} \mathbf{B}^{(m)} \right)$$

$$\mathbf{D}^{(m)} = \bigoplus_{\alpha=1}^m h_{m-\alpha}$$

$$\mathbf{B}^{(m)} = \bigoplus_{\alpha=1}^{M-m} h_{M-\alpha}$$

$$\mathbf{P}_{NM \times N}^{(3)} = \mathbf{1}_M \otimes \mathbf{I}_N$$

$$\mathbf{H}_{NM}^{(3)} = \text{diag}((\mathbf{I}_N \cdot h_0), (\mathbf{I}_N \cdot h_1), \dots, (\mathbf{I}_N \cdot h_{M-1}))$$

$$\mathbf{A}_{(N+M-1) \times NM}^{(3)} = \sum_{\alpha=0}^{M-1} \mathbf{A}_{(N+M-1) \times N}^{(\alpha)}$$

$$\mathbf{A}_{(N+M-1) \times N}^{(\alpha)} = \mathbf{I}_{N+M-1}^{(\alpha \rightarrow)} \cdot \mathbf{V}_{(N+M-1) \times N}$$

$$\mathbf{V}_{(N+M-1) \times N} = \mathbf{I}_N$$

Przykłady: Niech $N=5$, $M=3$.

$$\mathbf{X} = [x_0, x_1, x_2, x_3, x_4]^T$$

$$\mathbf{H} = [h_0, h_1, h_2]^T$$

$$\mathbf{Y} = \mathbf{A}_{7 \times 15}^{(j)} \cdot \mathbf{H}_{15}^{(j)} \cdot \mathbf{P}_{15 \times 5}^{(j)} \cdot \mathbf{X}$$

$$\mathbf{P}_{15 \times 5}^{(1)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{P}_{15 \times 5}^{(2)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ \hline 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{P}_{15 \times 5}^{(3)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ \hline 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ \hline 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{H}_{15}^{(1)} = \text{diag}(h_0, h_1, h_2, h_0, h_1, h_2, h_0, h_1, h_2, h_0, h_1, h_2, h_0, h_1, h_2),$$

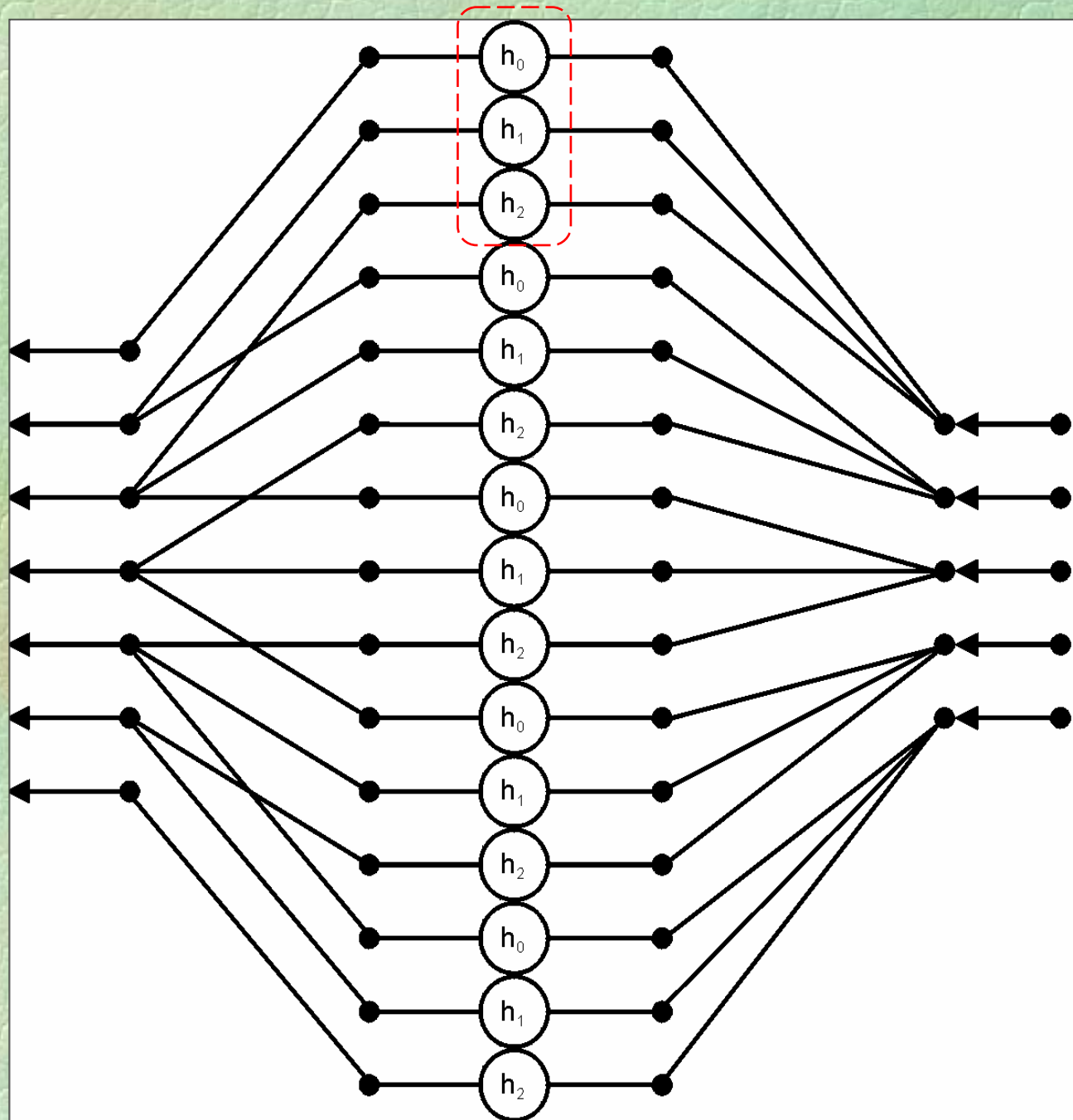
$$\mathbf{H}_{15}^{(2)} = \text{diag}(h_0, h_1, h_0, h_2, h_1, h_0, h_2, h_1, h_0, h_2, h_1, h_0, h_2, h_1, h_2),$$

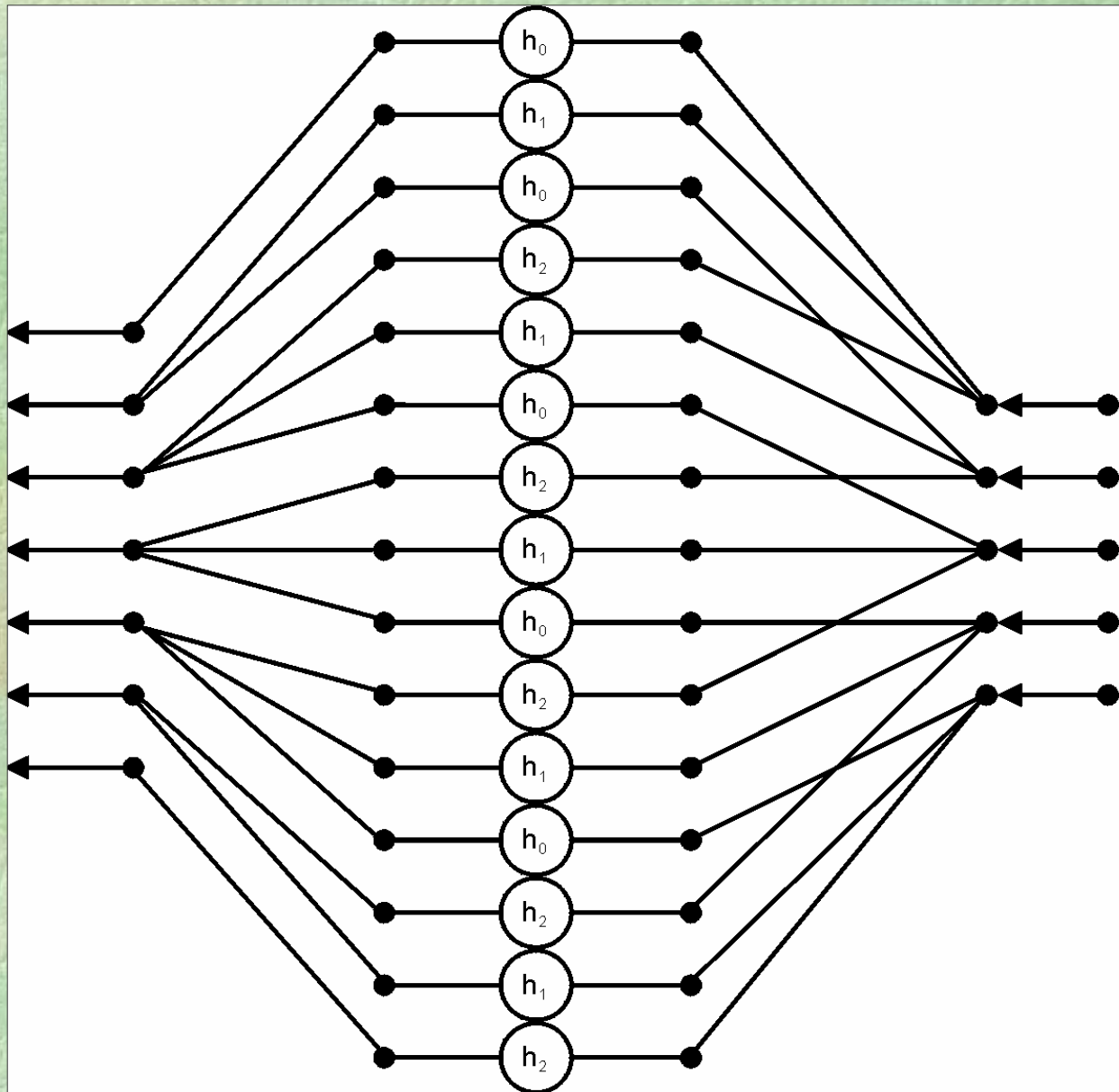
$$\mathbf{H}_{15}^{(3)} = \text{diag}(h_0, h_0, h_0, h_0, h_0, h_1, h_1, h_1, h_1, h_1, h_2, h_2, h_2, h_2, h_2);$$

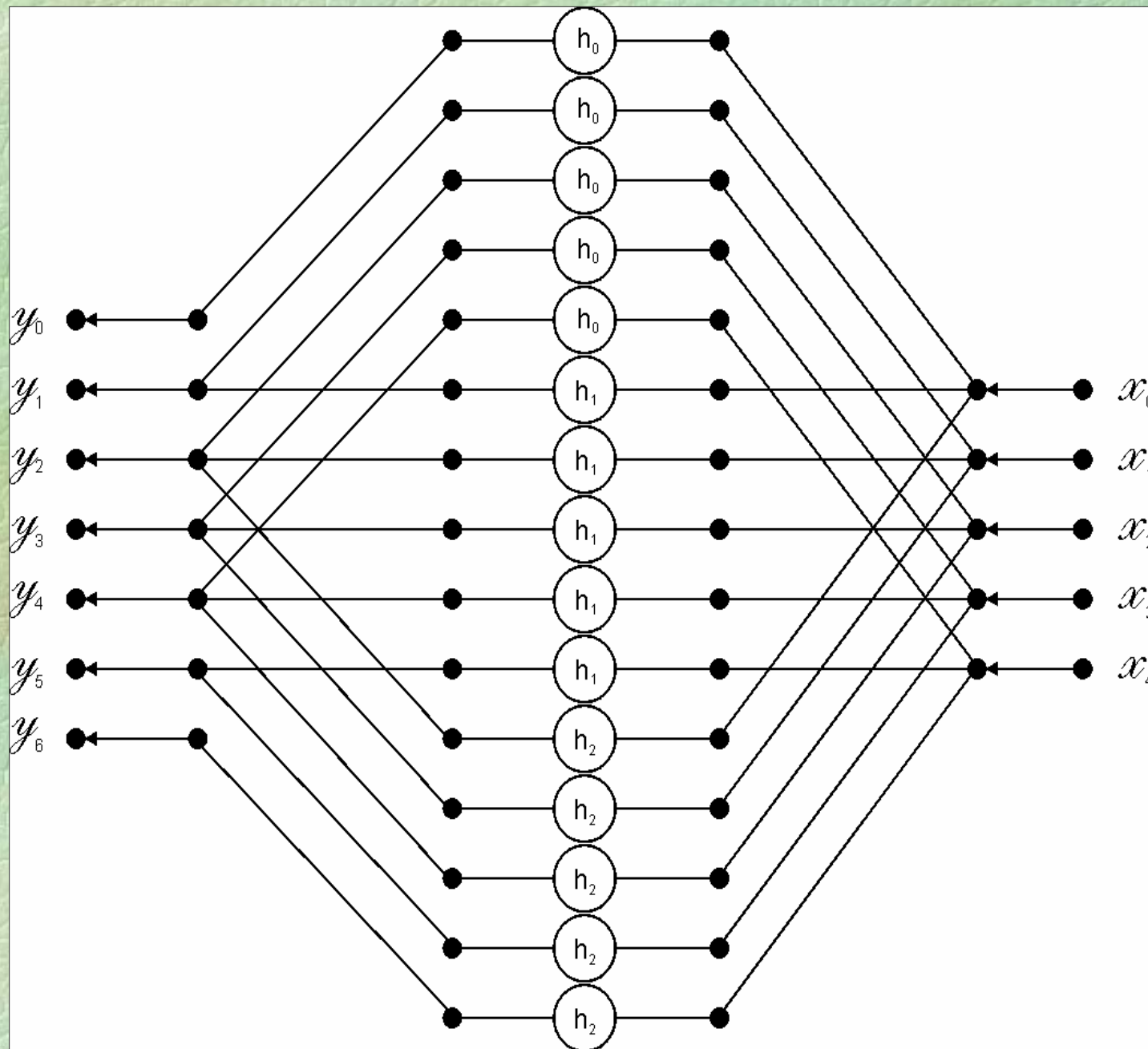
$$\mathbf{A}_{7 \times 15}^{(1)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{A}_{7 \times 15}^{(2)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{A}_{7 \times 15}^{(3)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$



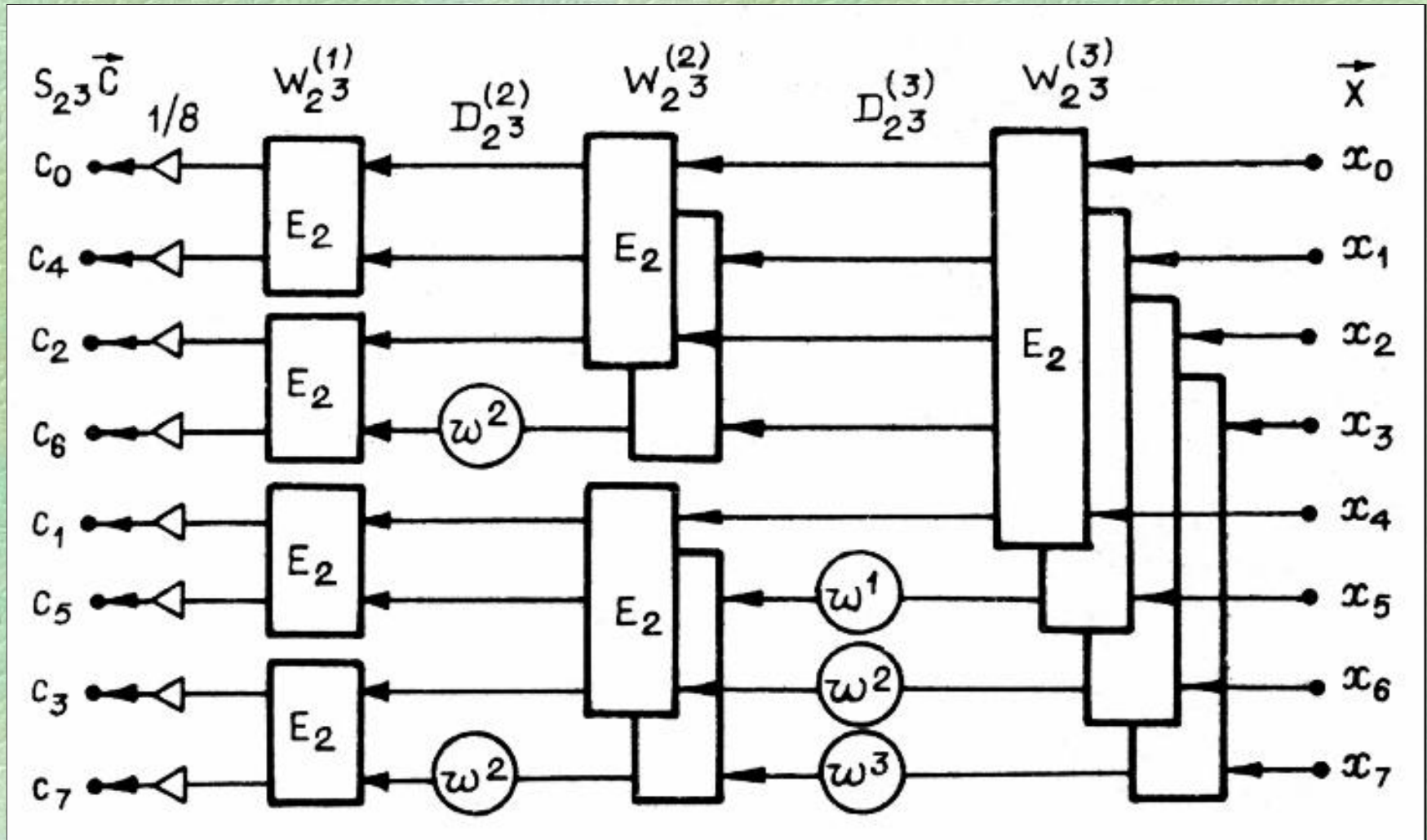




Równoległe algorytmy

FFT

Graf klasycznego algorytmu FFT dla $N=2^3$



$$\mathbf{X}_{r^g}^{(\alpha)} = [x_{\alpha M}, x_{\alpha M+1}, \dots, x_{(\alpha+1)M-1}]^T$$

$$\mathbf{C}_{r^m}^{(\alpha,g)} = [c_0^{(\alpha)}, c_1^{(\alpha)}, \dots, c_{N-1}^{(\alpha)}]^T$$

$$\mathbf{C}_{r^m}^{(\alpha,g)} = \mathbf{\Phi}_{r^m}^{(\alpha M \rightarrow)} (\mathbf{W}_{r^m}^{(m)} \cdot \mathbf{D}_{r^m}^{(m)} \cdot \dots \cdot \mathbf{W}_{r^m}^{(m-g+1)} \cdot \mathbf{D}_{r^m}^{(m-g+1)} \times \\ \times ((\mathbf{I}_{r^{m-g}} \otimes \mathbf{1}_M) \cdot \mathbf{S}_{r^g} \mathbf{X}_{r^g}^{(\alpha)}))$$

$$\mathbf{S}_{r^m}$$

$$s(k+1, (n)_r + 1) = 1$$

$$k = n = \overline{0, r^m - 1}$$

$$r = 2, 3, \dots$$

$$\mathbf{W}_{r^m}^{(i)} = \mathbf{I}_{r^{m-i}} \otimes \mathbf{E}_r \otimes \mathbf{I}_{r^i - 1}, \quad i = \overline{1, m}$$

$$\mathbf{D}_{r^m}^{(i)} = \mathbf{I}_{r^{m-i}} \otimes \mathbf{D}_{r^i}$$

$$\mathbf{D}_{r^i} = \text{diag}(\mathbf{D}_R^{(0)}, \mathbf{D}_R^{(1)}, \dots, \mathbf{D}_R^{(r-1)})$$

$$R = r^{i-1}$$

$$\mathbf{D}_R^{(k)} = \text{diag}(w^0, w^1, w^2, \dots, w^{(r-1)})^{(k)}$$

$$w = \exp(-j \frac{2\pi}{r^{(i)}})$$

$$\mathbf{\Phi}_{r^m}^{(\alpha M \rightarrow)} = \text{diag}(\omega^0, \omega^{\alpha M}, \omega^{2\alpha M}, \dots, \omega^{(N-1) \cdot \alpha M})$$

Wprowadźmy nowy superwektor o wymiarze $\frac{N^2}{M} \times 1$

$$\mathbf{C}_{r^{2m-g}}^{(1)} = [(\mathbf{C}_{r^m}^{(0,g)})^T, (\mathbf{C}_{r^m}^{(1,g)})^T, \dots, (\mathbf{C}_{r^m}^{((\frac{N}{M}-1),g)})^T]^T$$

$$\mathbf{C}_{r^m}^{(\alpha,g)}$$

$$\mathbf{C}_{r^{2m-g}}^{(1)} = \mathbf{F}_{r^{2m-g}} \cdot \mathbf{W}_{r^{2m-g}}^{(m)} \cdot \mathbf{D}_{r^{2m-g}}^{(m)} \cdot \dots \cdot \mathbf{W}_{r^{2m-g}}^{(m-g+1)} \cdot \mathbf{D}_{r^{2m-g}}^{(m-g+1)} \mathbf{X}_{r^{2m-g}}$$

$$\mathbf{X}_{r^{2m-g}}^{(1)} = ((\mathbf{I}_{\frac{N}{M}} \otimes \mathbf{I}_{r^g} \otimes \mathbf{1}_{\frac{N}{M}}) \cdot (\mathbf{I}_{\frac{N}{M}} \otimes \mathbf{S}_{r^g}) \cdot \mathbf{X}_{r^m})$$

$$\mathbf{F}_{r^{2m-g}} = \text{diag} [\mathbf{I}_{r^m}, \Phi_{r^m}^{(M \rightarrow)}, \Phi_{r^m}^{(2M \rightarrow)}, \dots, \Phi_{r^m}^{(\alpha M \rightarrow)}, \dots, \Phi_{r^m}^{((\frac{N}{M}-1)M \rightarrow)}]$$

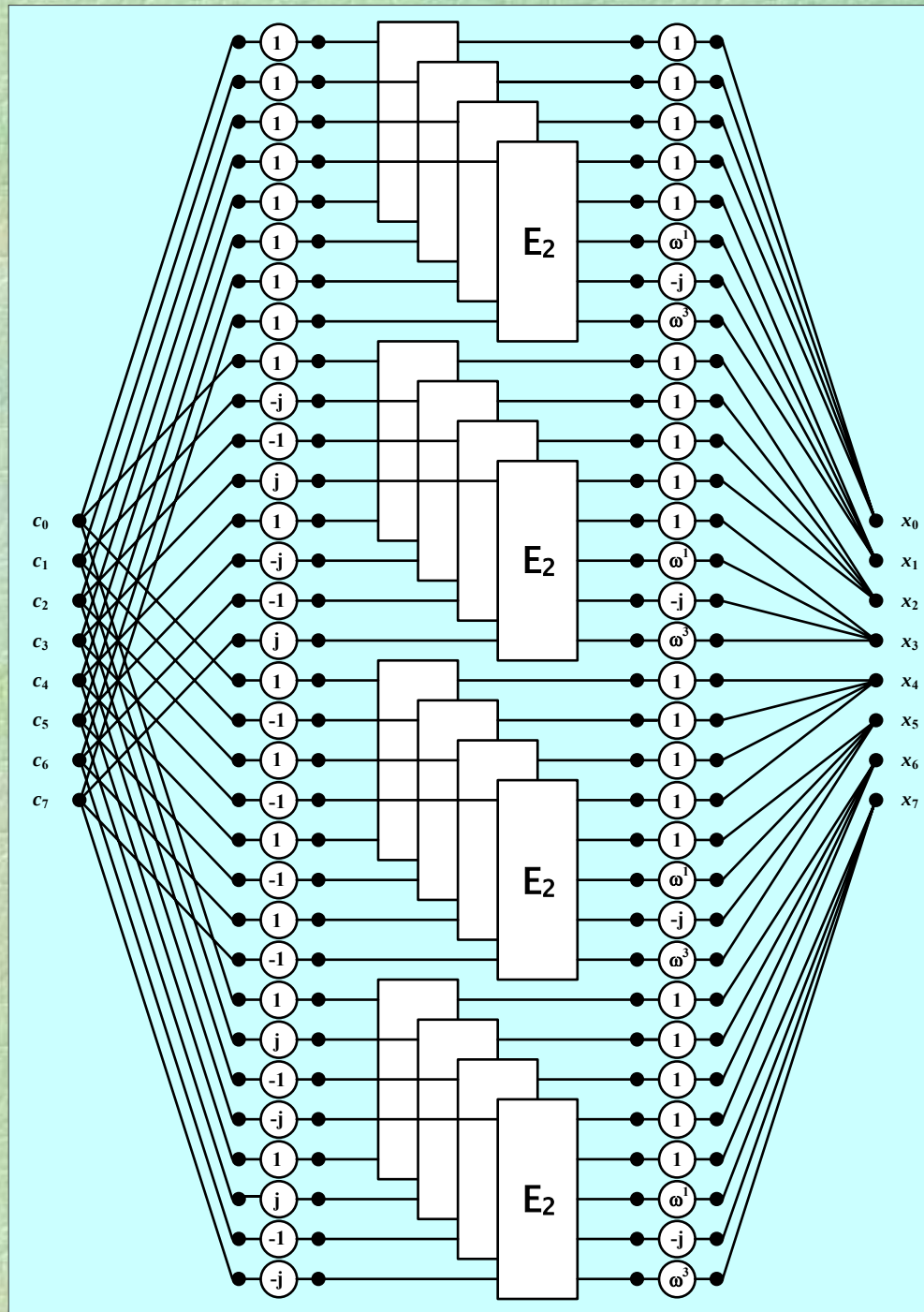
$$\mathbf{W}_{r^{2m-g}}^{(i)} = \mathbf{I}_{\frac{N}{M}} \otimes \mathbf{W}_{r^m}^{(i)}$$

$$\mathbf{D}_{r^{2m-g}}^{(i)} = \mathbf{I}_{\frac{N}{M}} \otimes \mathbf{D}_{r^m}^{(i)}$$

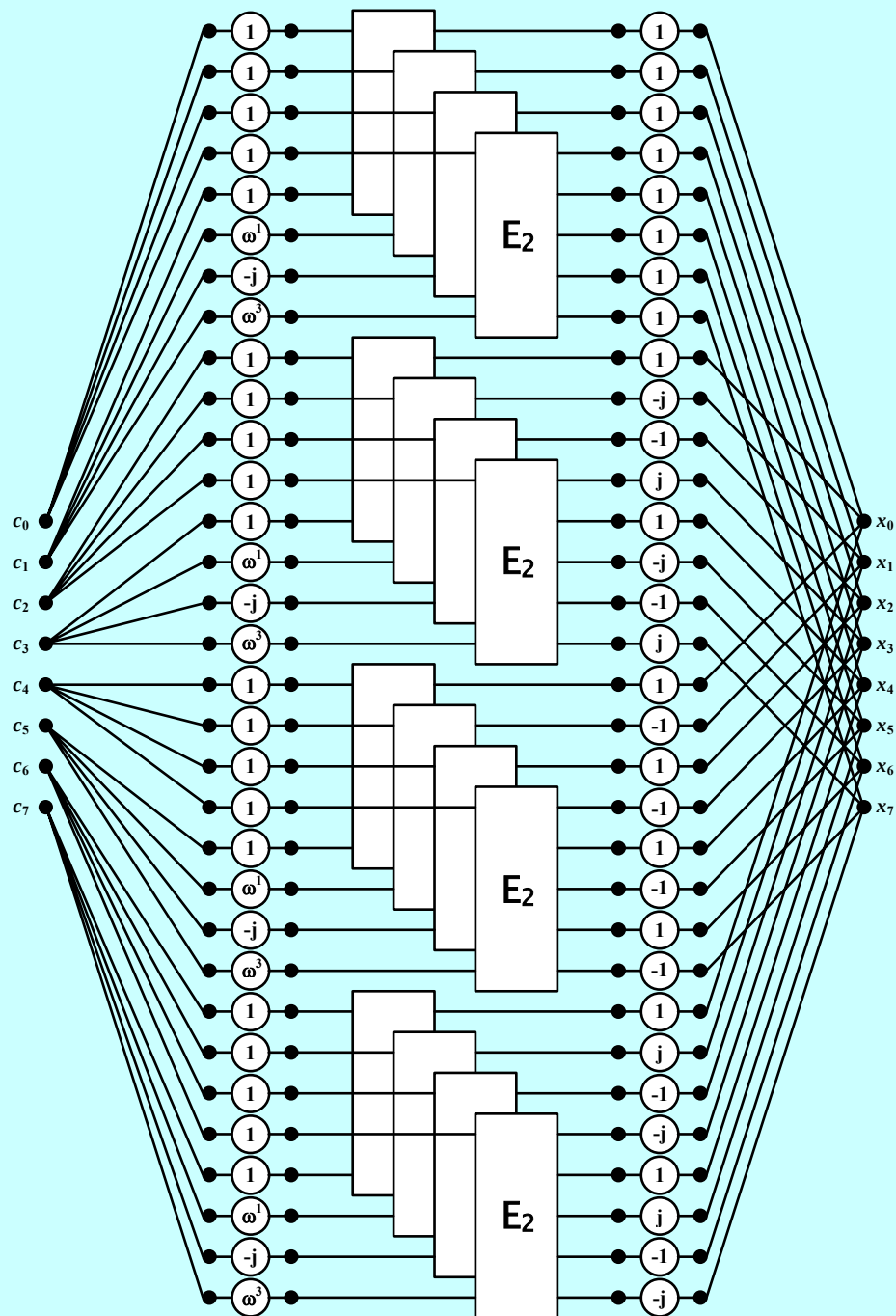
Równoległy
algorytm DFT w
bazie DEF z
podziałem w czasie
dla $N=8$, $M=2$

$$\mathbf{C}_{r^m} = \mathbf{A}_{r^m \times r^{2m-g}}^{(1)} \cdot \mathbf{C}_{r^{2m-g}}^{(1)}$$

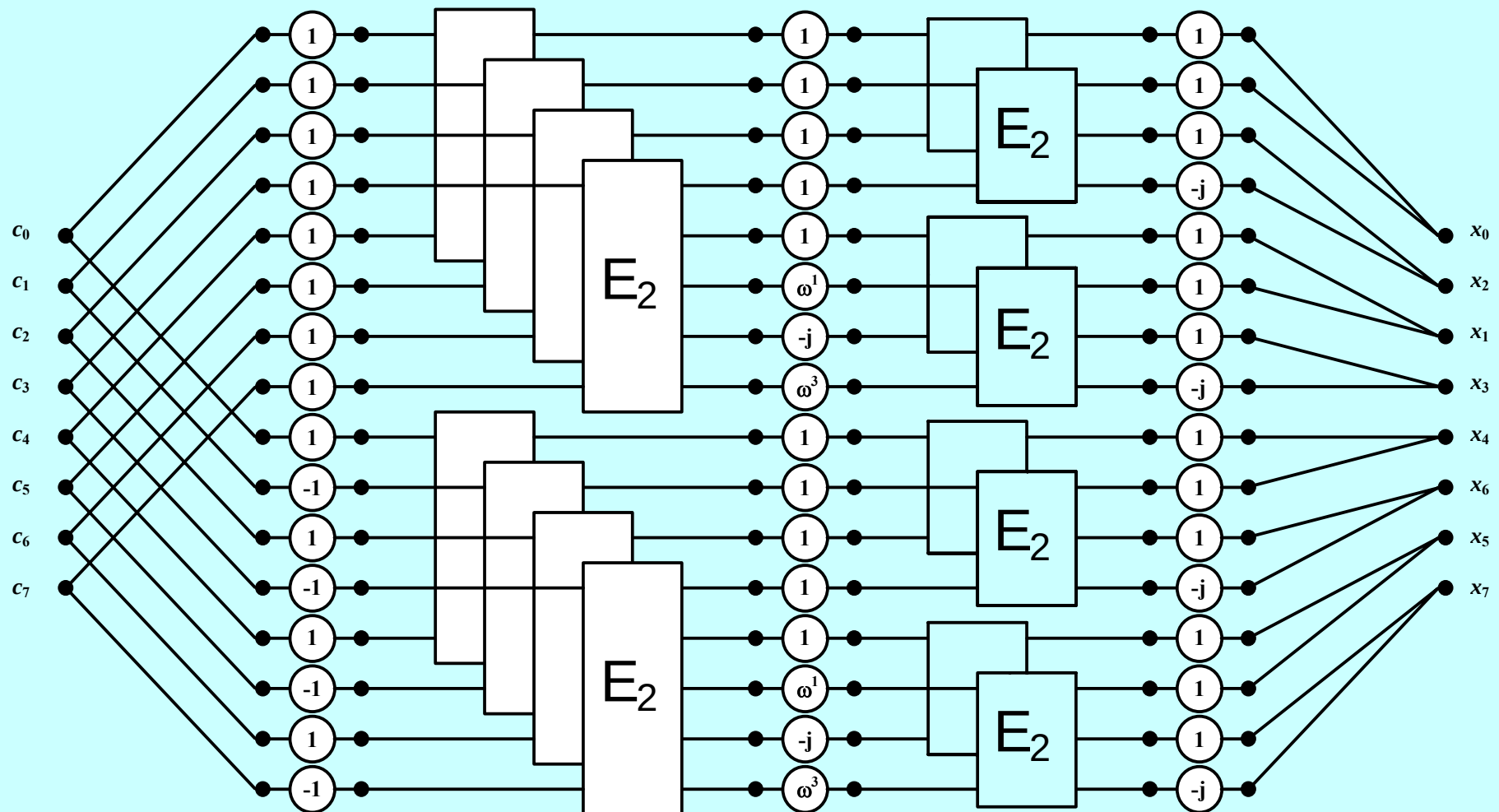
$$\mathbf{A}_{r^m \times r^{2m-g}}^{(1)} = \mathbf{1}_{\frac{N}{M}}^T \otimes \mathbf{I}_{r^m}$$



Równoległy
algorytm DFT
s podziałem w
częstotliwości
w bazie DEF
dla $N=8$, $M=2$



Równoległy algorytm DFT w bazie DEF z podziałem w czasie dla $N=8$, $M=2^2$



Równoległy algorytm DFT w bazie Wilensina-Krestensona dla $N=9$, $M=3$.

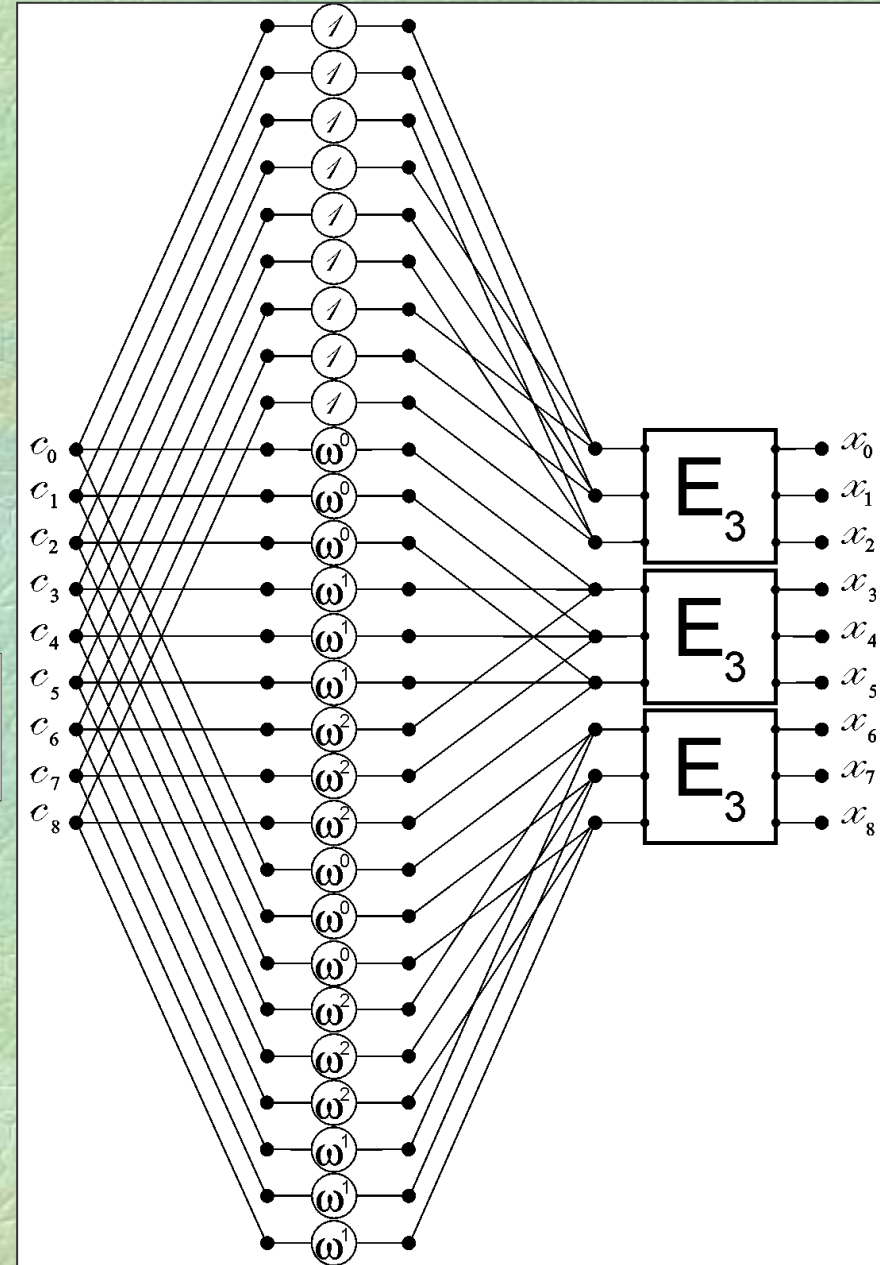
$$\mathbf{C}_{r^m} = \mathbf{A}_{r^m \times r^{2m-g}} \cdot \mathbf{C}_{r^{2m-g}},$$

$$\mathbf{C}_{r^{2m-g}}(t) = \Phi_{r^{2m-g}} (\mathbf{I}_{r^{m-g}} \otimes \mathbf{P}_{\frac{N}{M}} \otimes \mathbf{I}_M) \cdot$$

$$\cdot (\mathbf{T}_N^{(m)} \cdot \mathbf{T}_N^{(m-1)} \cdot \dots \cdot \mathbf{T}_N^{(m-g+1)}) \mathbf{X}_{r^m},$$

$$\mathbf{T}_N^{(i)} = \mathbf{I}_{r^{m-g}} \otimes \mathbf{V}_M^{(i)};$$

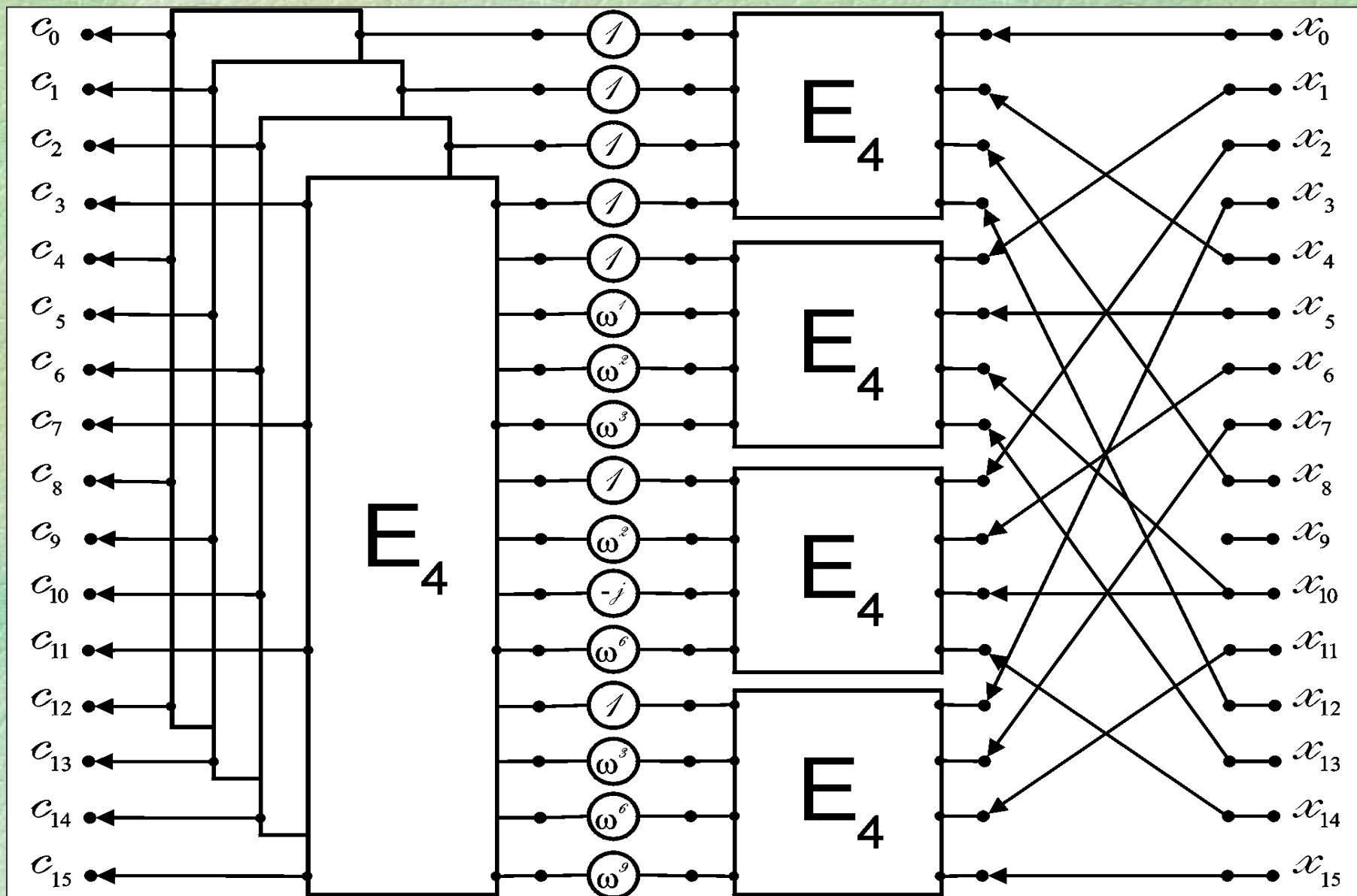
$$\mathbf{V}_M^{(i)} = \mathbf{I}_{r^{g-i}} \otimes \mathbf{E}_r \otimes \mathbf{I}_{r^{i-1}}$$

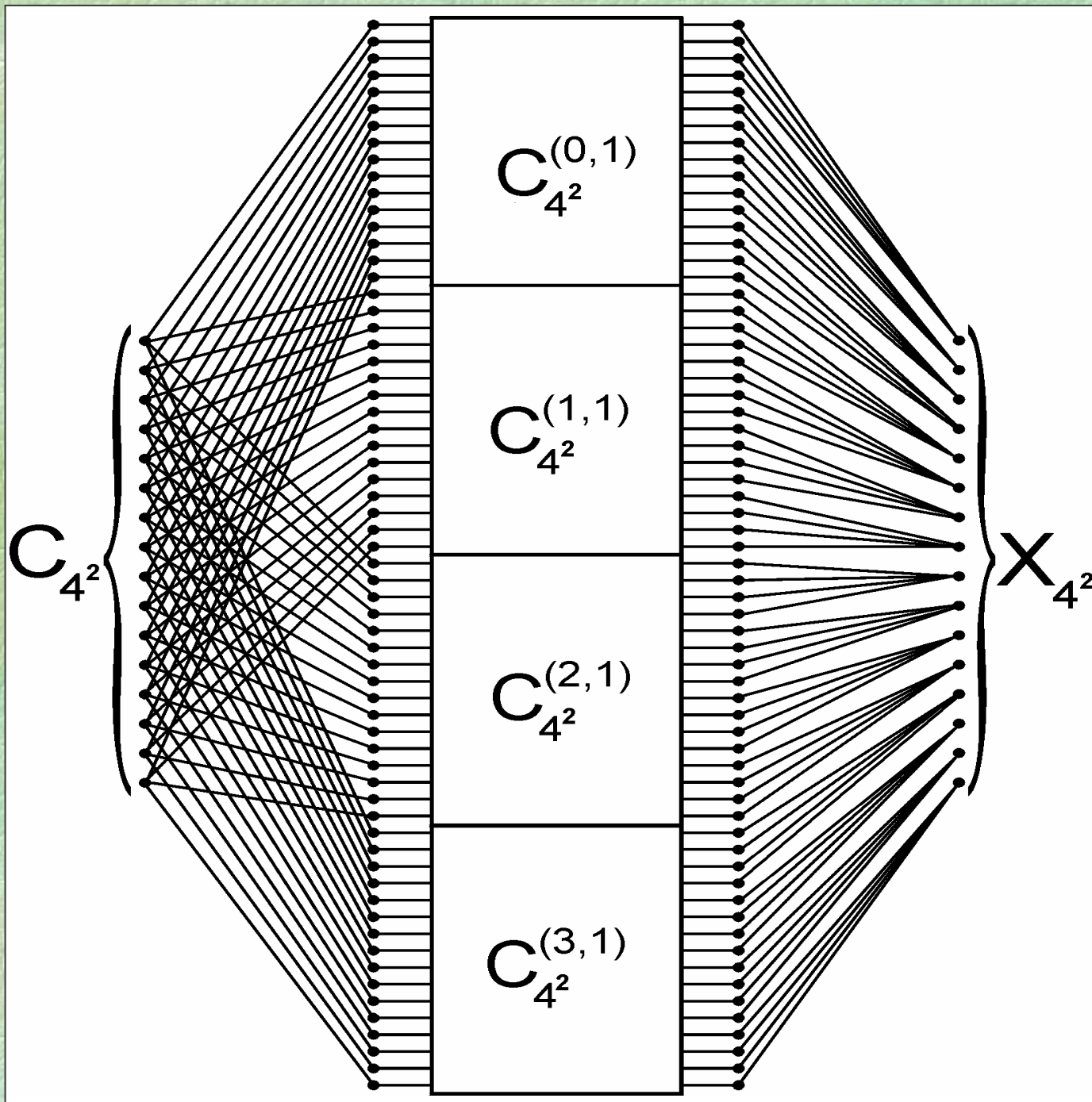


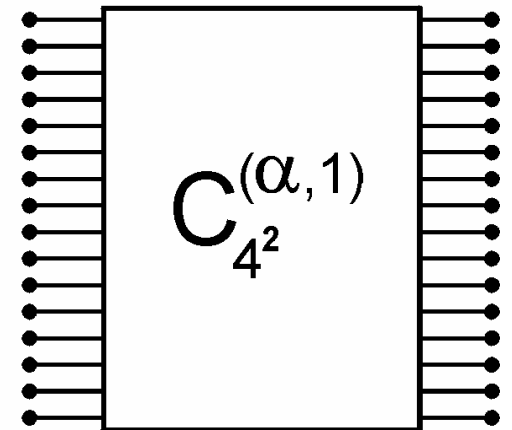
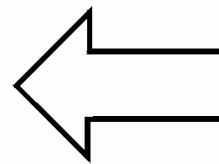
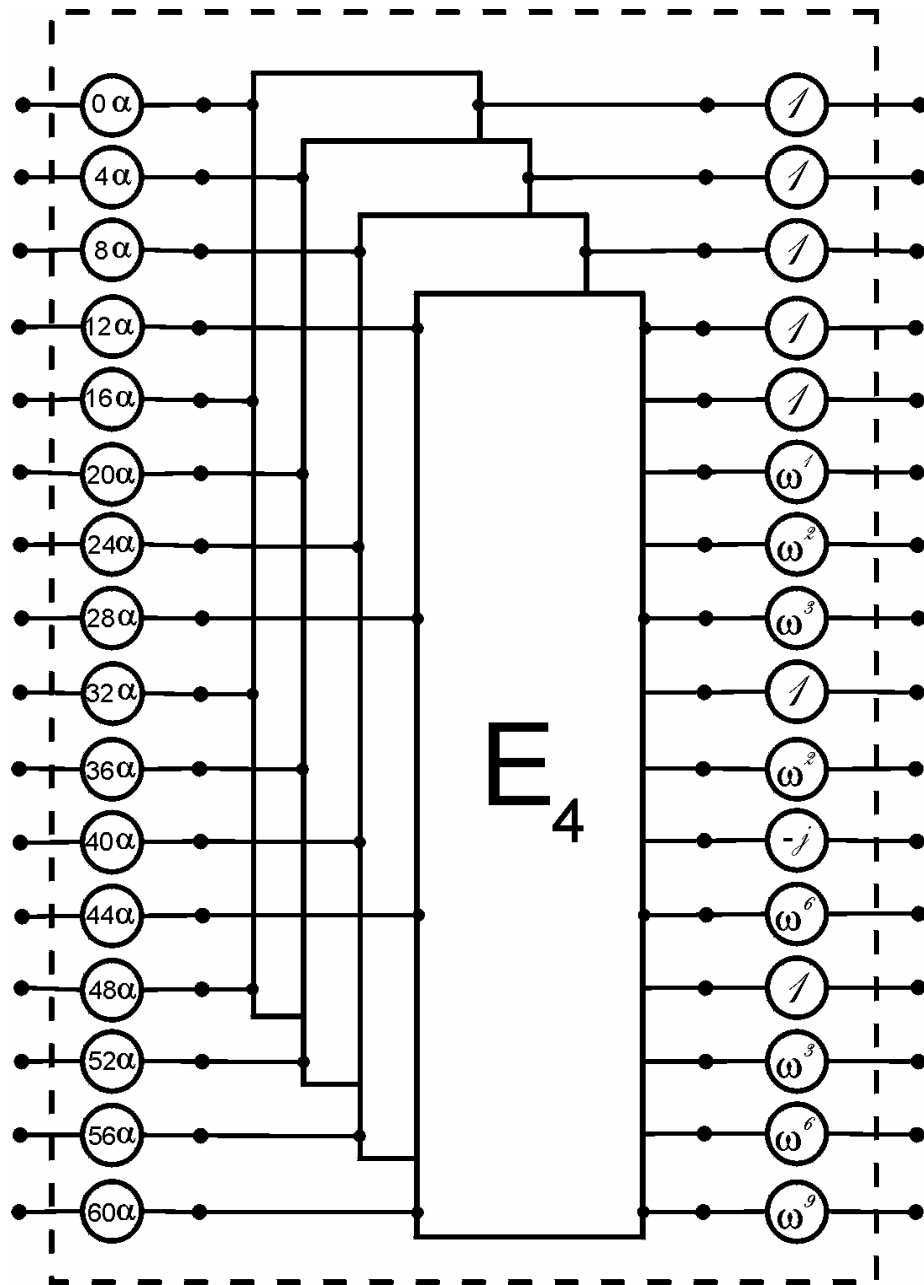
$$N = (...((r_0^{m_0})^{m_1})^{m_2} \dots)^{m_\varphi}$$

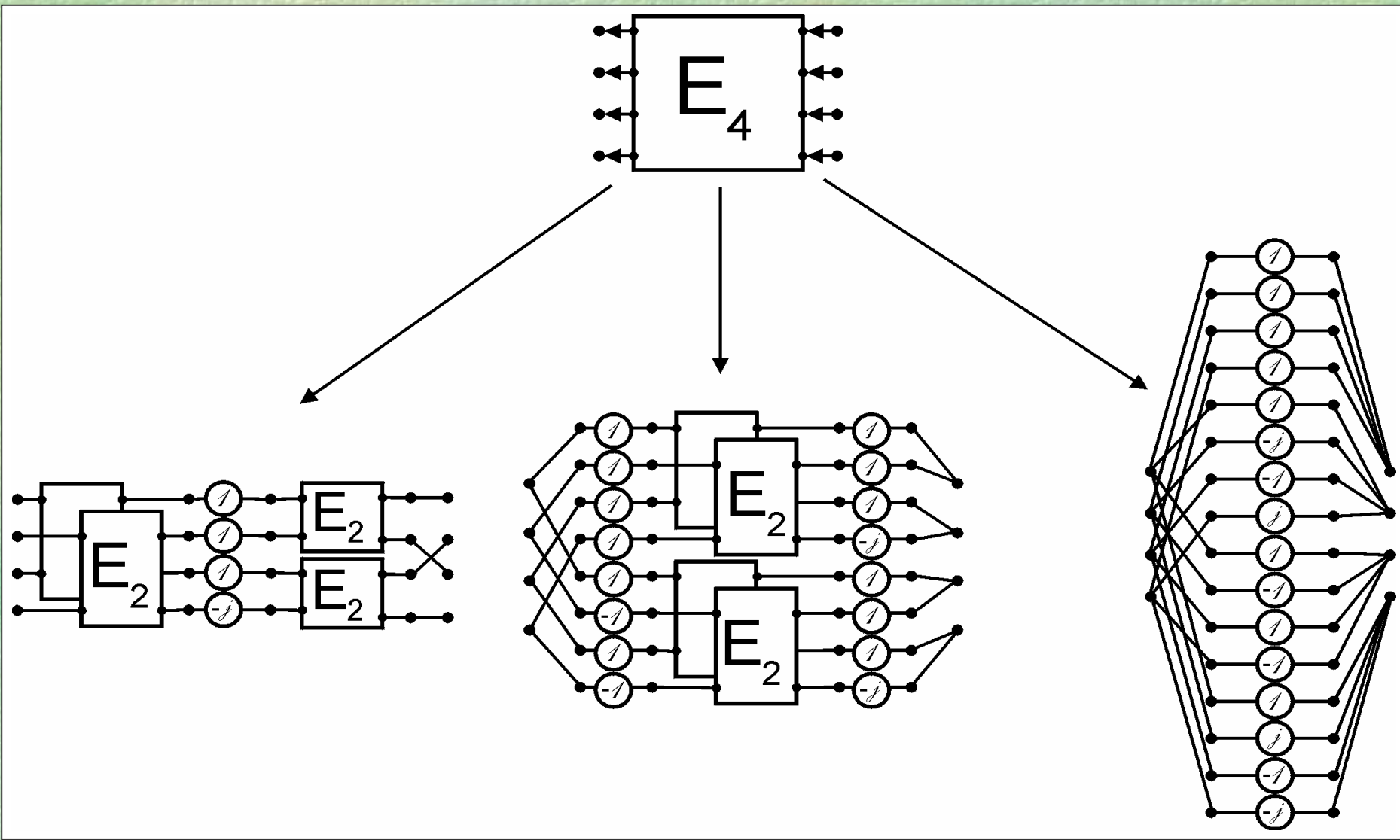
$$N = r_0^{\prod_{l=0}^{\varphi} m_l}$$

$$r_0, m_0 \div m_\varphi$$









Równoległe algorytmy DWT

Dyskretna transformata falkowa jest mocnym instrumentem współczesnej cyfrowej obróbki sygnałów.

Jej stosowanie jest skuteczne zwłaszcza przy analizie sygnałów niestacjonarnych, gdy użycie własności transformaty Fouriera daje niedokładne rezultaty.

Istotą DWT (ang. *Discrete Wavelet Transform*) jest podział pierwotnego sygnału za pomocą pary filtrów (górnoprzepustowego i dolnoprzepustowego) na dwie składowe, następnie dekompozycja hierarchiczna jednej lub obydwu składowych w ten sam sposób.

Takie podejście zostało nazwane piramidą Mallata lub szybką transformatą falkową.

W rzeczywistości na każdym etapie realizacji piramidy Mallata jest wykonana bazowa procedura splotu wejściowego dla tego etapu szeregu danych z dwoma zestawami współczynników wagowych:

$$a_n^{(i)} = \sum_{m=0}^{M_L - 1} g_m \cdot a_{2n-m}^{(i-1)}, \quad i = \overline{1, K},$$

$$d_n^{(i)} = \sum_{m=0}^{M_H - 1} h_m \cdot a_{2n-m}^{(i-1)}, \quad a_n^{(0)} = x(n)$$

oraz operacja „downsamplingu” – odrzucania co drugiej próbki odfiltrowanego sygnału w każdej ze składowych w celu redukcji nadmiarowości zbioru danych.

Realizacja powyższej procedury wymaga wykonania NL operacji mnożenia i prawie tyle samo operacji dodawania.

W przypadku dużych N czas realizacji całkowitego rozkładu sygnału jest dość istotny.

Z tego powodu celem jest poszukiwanie metod i rozwiązań minimalizacji tego czasu.

Jednym z racjonalnych sposobów zwiększenia wydajności obliczeń transformaty falkowej może być włączenie specjalizowanych procesorów z równoległą obróbką danych w skład sprzętu do przetwarzania sygnałów.

Jednak wykonana analiza wskazuje na to, że w znanych rozwiązaniach nie wykorzystano wszystkich możliwości zrównoleglenia obliczeń.

Fakt ten jest spowodowany tym, że we wszystkich znanych rozwiązaniach wymiar wektora jednocześnie przetwarzanych danych nie przekracza wielkości $N/2^k$ dla dowolnego k -tego etapu algorytmu.

Podkreślony fakt nie pozwala na zwiększenie prędkości obliczeń poza pewną wielkość, niezależnie od zwiększenia zasobów sprzętowych procesora.

Z tego powodu zastało rozpatrzone podejście do organizacji zrównoleglenia obliczeń przy implementacji dowolnej iteracji piramidy Mallata, pozwalające wyjść poza ramy wspomnianego rozmiaru i przez to zminimalizować czas oczekiwania na końcowy rezultat, czyli zwiększyć prędkość obliczeń.

Niech $\mathbf{X} = [x_0, x_1, \dots, x_{\frac{N}{2^k} + L - 3}]^T$

- wektor-kolumna danych wejściowych, a

$\mathbf{Y} = [y_0, y_1, \dots, y_{\frac{N}{2^k} - 1}]^T$ - wektor-kolumnowy

danych wyjściowych na k -ej iteracji piramidy Mallata,

$k = \overline{0, K-1}$ - numer kroku dekompozycji określający poziom detalizacji analizy falkowej,

K - kompletna ilość kroków dekompozycji.

Wprowadźmy macierz

$$\mathbf{F}_{2 \times L} = \begin{bmatrix} h_0 & h_1 & \cdots & h_{L-1} \\ g_0 & g_1 & \cdots & g_{L-1} \end{bmatrix}$$

o rozmiarze $(2 \times L)$, której elementy reprezentują odpowiednio współczynniki filtra dolnoprzepustowego i górnoprzepustowego.

Będziemy nazywać tę macierz „**macierz bazy DWT**”, natomiast operację mnożenia tej macierzy przez wektor odpowiednich danych nazwiemy „**bazową operacją DWT**”.

Zbudujemy teraz trzy macierze wspomagające:

★ macierz superwektoryzacji danych

$$\mathbf{P}_{\frac{NL}{2^{k+1}} \times (\frac{N}{2^k} + L - 2)} = \begin{matrix} \frac{N}{2^{k+1}} - 1 \\ \blacksquare \\ \hline \blacksquare \\ i=0 \end{matrix} \mathbf{P}_{L \times (\frac{N}{2^k} + L - 2)} \cdot \mathbf{I}_{\frac{N}{2^k} + L - 2}^{(2i \rightarrow)}$$

$$\mathbf{P}_{L \times (\frac{N}{2^k} + L - 2)} = \mathbf{I}_L \blacksquare \blacksquare \mathbf{0}_{L \times (\frac{N}{2^k} - 2)}$$

gdzie $\mathbf{I}_L$ - tu i w dalszej części wykładu macierz jednostkowa stopnia L ,

$\mathbf{0}_{L \times (\frac{N}{2^k} - 2)}$ - macierz zerowa o wymiarze określonym za pomocą dolnego indeksu,

★ macierz filtracji

$$\tilde{\mathbf{F}}_{\frac{N}{2^k} \times \frac{NL}{2^{k+1}}} = \mathbf{I}_{\frac{N}{2^{k+1}}} \otimes \mathbf{F}_{2 \times L}$$

gdzie $\otimes$ -symbol iloczynu Kroneckera;

★ macierz przesyłu danych

$$\mathbf{P}_{\frac{N}{2^k}} = \left(\begin{array}{c} \frac{N}{2^{k+1}} - 1 \\ \blacksquare \\ \frac{N}{2^k} \\ \blacksquare \\ i=0 \end{array} e_{2i} \right) \blacksquare \left(\begin{array}{c} \frac{N}{2^{k+1}} - 1 \\ \blacksquare \\ \frac{N}{2^k} \\ \blacksquare \\ i=0 \end{array} e_{2i+1} \right)$$

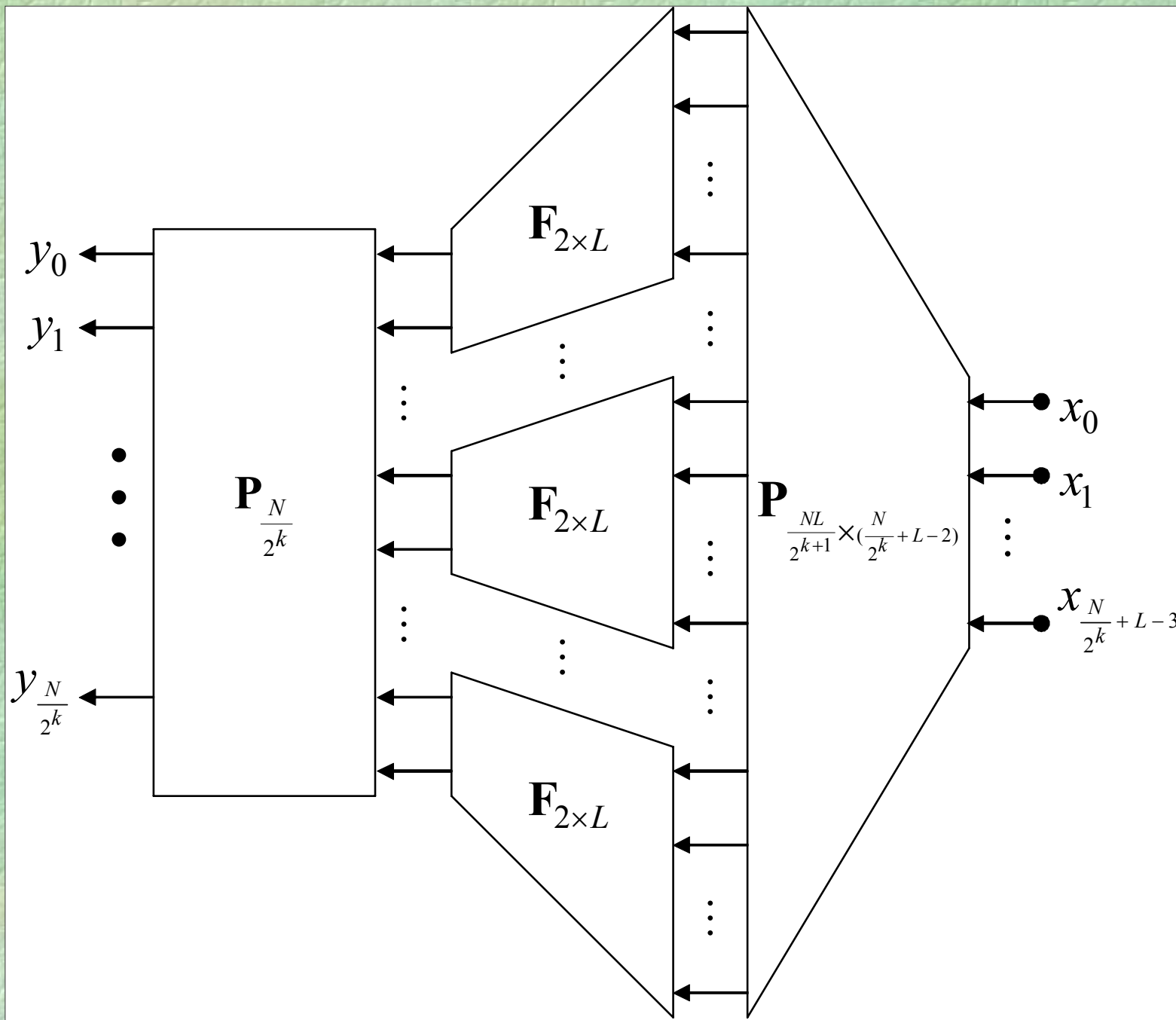
gdzie e_n - wiersz macierzy jednostkowej rzędu N z numerem określonym za pomocą dolnego indeksu.

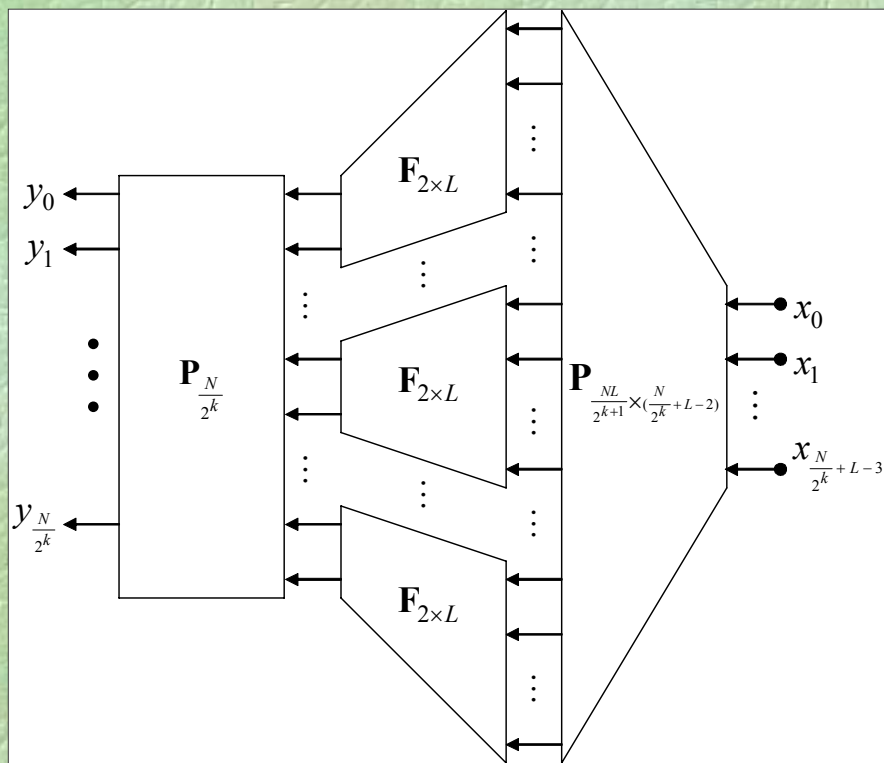
Wtedy nowy równoległy algorytm implementacji obliczeń dla jednej, dowolnej iteracji piramidy Mallata można opisać za pomocą następującej procedury wektorowo-macierzowej:

$$\mathbf{Y}_{N \times 1} = \mathbf{P}_{\frac{N}{2^k}} \cdot \tilde{\mathbf{F}}_{\frac{N}{2^k} \times \frac{NL}{2^{k+1}}} \cdot \mathbf{P}_{\frac{NL}{2^{k+1}} \times (\frac{N}{2^k} + L - 2)} \cdot \mathbf{X}_{(\frac{N}{2^k} + L - 2) \times 1}$$

gdzie $\mathbf{X}_{(\frac{N}{2^k} + L - 2) \times 1} = [x_0, x_1, \dots, x_{N-1}, x_N, \dots, x_{\frac{N}{2^k} + L - 3}]^T$

- wektor-kolumnowy danych wejściowych dla każdej iteracji, uzupełniony według metody Mallata $(L-2)$ dodatkowymi elementami.

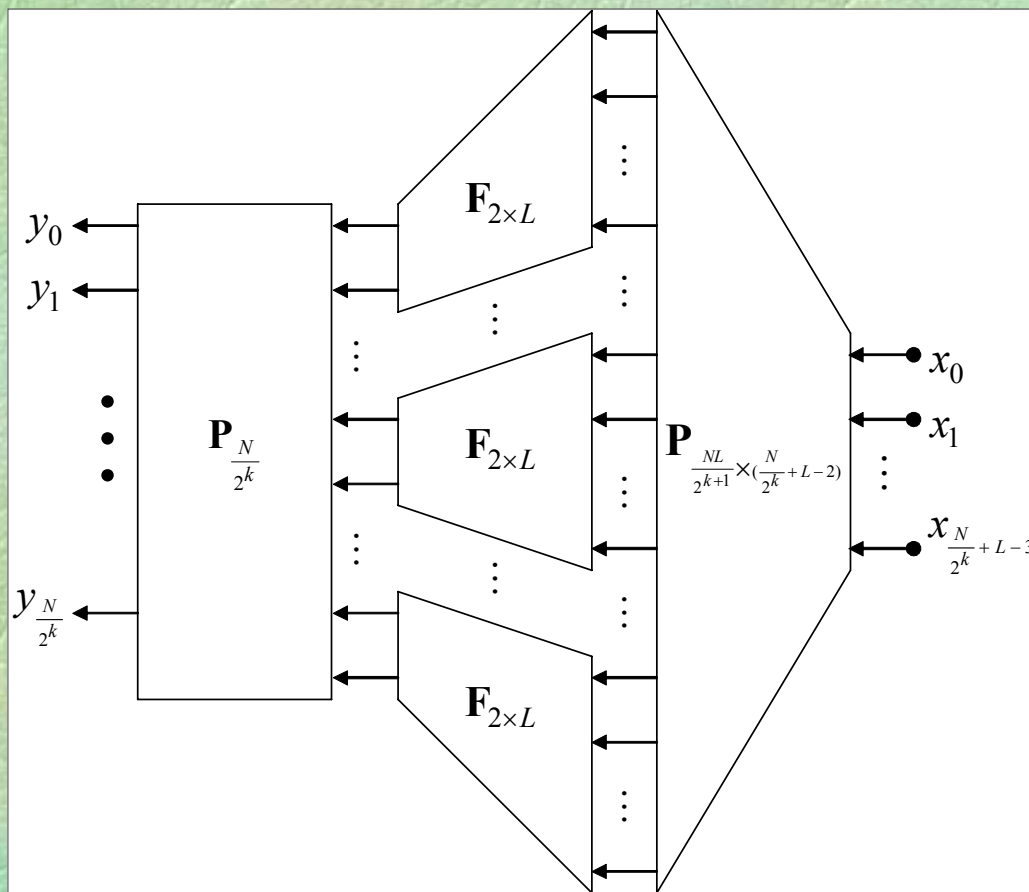




Jak to łatwo zauważyć proces obliczeniowy może teraz być podzielony na niezależnych wątków, w każdym z których wykonuje się bazowa operacja DWT.

Opisane podejście pozwala zrównoleglić proces implementacji bazowej procedury DWT w przestrzeni (zwektoryzować) i przez to zminimalizować opóźnienie końcowych rezultatów.

$$\tau = \tau_{SW} + \tau_B + \tau_P$$



$$\tau = \tau_{SW} + \tau_B + \tau_P$$

Jak to widać, ostateczna ocena czasu wykonania całej procedury DWT na k -tej iteracji za pomocą zaproponowanego podejścia zrównoleglenia obliczeń w znacznym stopniu będzie zależała od sposobu realizacji operacji bazowej.

Obliczenie **bazowej operacji DWT** wymaga wykonania **$2L$** mnożeń i **$2(L-1)$** dodawań, a cała procedura „szybkiego” DWT – blisko **$2NL$** .

Na dzień dzisiejszy nie istnieją rozwiązania pozwalające zredukować liczbę operacji arytmetycznych przy realizacji bazowej procedury DWT w celu jeszcze większego zmniejszenia tej wartości.

Okazuje się, że taka możliwość jednak istnieje.

Uwzględniając cechę szczególną macierzy bazy DWT a mianowicie to, że jej elementy są liczbami stałymi, można skonstruować algorytmy realizacji bazowej procedury DWT ze zredukowaną ilością mnożeń.

Tu zostało pokazane jedno z możliwych podejść, pozwalające zredukować liczbę układów mnożących względem znanych rozwiązań.

Zbudujemy kilka macierzy wspomagających:

- macierz superwektoryzacji danych wejściowych

$$\mathbf{P}_{\frac{NL}{2^{k+1}} \times (\frac{N}{2^k} + L - 2)} = \begin{matrix} \frac{N}{2^{k+1}} - 1 \\ \blacksquare \\ \hline \blacksquare \\ i=0 \end{matrix} \mathbf{P}_{L \times (\frac{N}{2^k} + L - 2)} \cdot \mathbf{I}_{\frac{N}{2^k} + L - 2}^{(2i \rightarrow)}$$

$$\mathbf{P}_{L \times (\frac{N}{2^k} + L - 2)} = \mathbf{I}_L \blacksquare \blacksquare \mathbf{0}_{L \times (\frac{N}{2^k} - 2)}$$

- wektor-kolumnowy „współczynników wagowych” filtracji

$$\mathbf{F}_{\left[\frac{N}{2^k}(L+1)+L-2\right] \times 1} = \left[\mathbf{G}_{\frac{NL}{2^{k+1}} \times 1}, \mathbf{0}_{\left(\frac{N}{2^k}+L-2\right) \times 1}, \mathbf{H}_{\frac{NL}{2^{k+1}} \times 1} \right]^T$$

$$\mathbf{G}_{\frac{NL}{2^{k+1}}} = \mathbf{1}_{\frac{N}{2^{k+1}} \times 1} \otimes \left(\mathbf{I}_{\frac{L}{2}} \otimes \mathbf{J}_2 \right) \cdot \mathbf{G}_{L \times 1}$$

$$\mathbf{H}_{\frac{NL}{2^{k+1}}} = \mathbf{1}_{\frac{N}{2^{k+1}} \times 1} \otimes \left(\mathbf{I}_{\frac{L}{2}} \otimes \mathbf{J}_2 \right) \cdot \mathbf{H}_{L \times 1}$$

$$\mathbf{G}_{L \times 1} = [g_0, g_1, \dots, g_{L-1}]^T$$

-wektor-kolumna, opisujący współczynniki wagowe filtru dolnoprzepustowego,

$$\mathbf{H}_{L \times 1} = [h_0, h_1, \dots, h_{L-1}]^T$$

-wektor-kolumna, opisujący współczynniki wagowe filtru górnoprzepustowego;

- macierz „maskowania” mnożeń

$$\mathbf{B}_{\left[\frac{N}{2^{k+1}}(L+1)+\frac{L}{2}-1\right] \times \left[\frac{N}{2^k}(L+1)+L-2\right]} = \mathbf{I}_{\frac{N}{2^{k+1}}(L+1)+\frac{L}{2}-1} \otimes \mathbf{1}_{1 \times 2}$$

- macierz sumowania iloczynów

$$\mathbf{A}_{\frac{NL}{2^{k+2}}(\frac{L}{2}+1) \times \frac{N}{2^{k+1}}+L-1} = \mathbf{A}_{\frac{LN}{2^{k+3}} \times \frac{LN}{2^{k+2}}} \oplus \mathbf{A}_{\frac{N}{2^{k+1}} \times \frac{1}{2}(\frac{N}{2^k}+L-2)} \oplus \mathbf{A}_{\frac{LN}{2^{k+3}} \times \frac{LN}{2^{k+2}}}$$

$$\mathbf{A}_{\frac{LN}{2^{k+3}} \times \frac{LN}{2^{k+2}}} = \mathbf{I}_{\frac{LN}{2^{k+3}}} \otimes \mathbf{1}_{1 \times 2}$$

$$\mathbf{A}_{\frac{N}{2^{k+1}} \times \frac{1}{2}(\frac{N}{2^k}+L-2)} = \sum_{i=0}^{\frac{N}{2^{k+1}}-1} (\mathbf{1}_{1 \times \frac{L}{2}} \otimes \mathbf{0}_{1 \times [\frac{1}{2}(\frac{N}{2^k}+L-2)-L]}) \cdot \mathbf{I}_{\frac{1}{2}(\frac{N}{2^k}+L-2)}^{(i \rightarrow)}$$

- macierz „rozproszenia” danych

$$\mathbf{R}_{\frac{2N}{2^k} \times \frac{3N}{2^k}} = \mathbf{I}_{\frac{N}{2^k}} \oplus (\mathbf{1}_{2 \times 1} \otimes \mathbf{I}_{\frac{N}{2^k}}) \oplus \mathbf{I}_{\frac{N}{2^k}}$$

- wektor-kolumną „korekcji wag”

$$\tilde{\mathbf{F}}_{\frac{2N}{2^k} \times 1} = [\mathbf{0}_{\frac{N}{2^{k+1}} \times 1}, \tilde{\mathbf{G}}_{\frac{N}{2^{k+1}} \times 1}, \tilde{\mathbf{H}}_{\frac{N}{2^{k+1}} \times 1}, \mathbf{0}_{\frac{N}{2^{k+1}} \times 1}]^T, \text{ gdzie}$$

$$\tilde{\mathbf{G}}_{\frac{N}{2^{k+1}} \times 1} = [g(L), g(L), \dots, g(L)]^T$$

$$\tilde{\mathbf{H}}_{\frac{N}{2^{k+1}} \times 1} = [h(L), h(L), \dots, h(L)]^T$$

$$g(L) = \sum_{i=0}^{\frac{L}{2}-1} g_{2i} \cdot g_{2i+1}$$

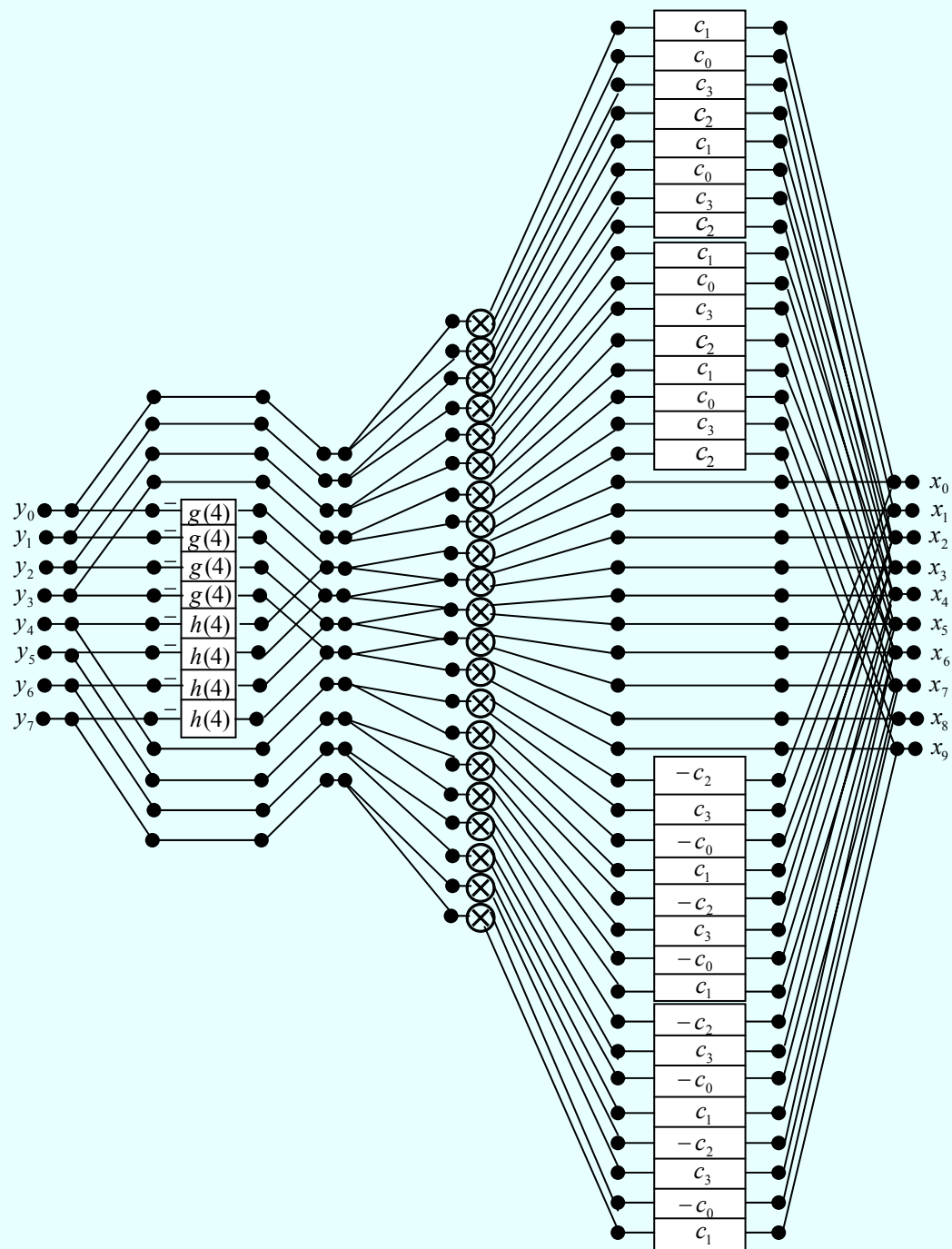
$$h(L) = \sum_{i=0}^{\frac{L}{2}-1} h_{2i} \cdot h_{2i+1}$$

Uwzględniając wprowadzone konstrukcje wektorowo-macierzowe, algorytmiczną strukturą procesora dla implementacji bazowej procedury DWT ze zredukowaną liczbą operacji mnożenia można przedstawić następująco:

$$\begin{aligned}
 \mathbf{Y} = & \mathbf{A}_{\frac{2N}{2^{k+1}} \times \frac{2N}{2^k}} \cdot (\tilde{\mathbf{F}}_{\frac{2N}{2^k} \times 1} \cdot (\mathbf{R}_{\frac{2N}{2^k} \times \frac{3N}{2^k}} \cdot (\mathbf{A}_{\frac{NL}{2^{k+2}} (\frac{L}{2} + 1) \times \frac{N}{2^{k+1}} + L - 1} \times \\
 & \times (\mathbf{B}_{[\frac{N}{2^{k+1}} (L+1) + \frac{L}{2} - 1] \times [\frac{N}{2^k} (L+1) + L - 2]} \circ (\mathbf{F}_{[\frac{N}{2^k} (L+1) + L - 2] \times 1} + \mathbf{P}_{\frac{NL}{2^{k+1}} \times (\frac{N}{2^k} + L - 2})_{3L \times L} \cdot \mathbf{X}))))))
 \end{aligned}$$

gdzie

$$\mathbf{A}_{\frac{2N}{2^{k+1}} \times \frac{2N}{2^k}} = ([1, -1] \otimes \mathbf{I}_{\frac{N}{2^k}}) \oplus ([-1, 1] \otimes \mathbf{I}_{\frac{N}{2^k}})$$



Zaproponowane podejście pozwala zredukować liczbę operacji mnożenia (lub układów mnożących w przypadku implementacji sprzętowej) z $N \cdot L$ do $\frac{1}{2}[N(L+1)+L-2]$ względem klasycznej metody kosztem powstania pewnej liczby dodatkowych operacji sumowania ze stałą.

Operacja ta jest jednak mniej skomplikowana niż operacja mnożenia, co w sumie daje zysk przy implementacji.

Wielkość tego zysku nie jest rzeczą trywialną i wymaga dalszych badań. Przeważnie jest to związane z dużą liczą wariantów organizacji obliczeń w ramach bloków i modułów tej struktury.