

1. **Historia rozwoju DSP. Jean Baptiste Joseph de Fourier., J.W.Cooley & J.W. Tukey. Główna cecha systemów DSP. Architektura komputerowa w aspekcie DSP. Typowe operacje DSP. Układy scalone dla DSP. Historia rozwoju procesorów sygnałowych.**

Historia rozwoju DSP:

- większość aparatu matematycznego i algorytmów opracowano już w latach 50 20go wieku
- podstawowym wymogiem był **czas przetwarzania**, czyli możliwość dokonania przewidzianej symulacji w **dopuszczalnym** dla danego zastosowania czasie
- Wcześniejsze systemy były zdolne jedynie do zapisania przebiegu poszczególnych sygnałów w pamięci komputera i przetworzenia ich później.
- Takie systemy nie mogły mieć wpływu na przebieg badanego procesu i nie mogły podjąć natychmiastowej decyzji na podstawie zgromadzonych danych

Jean Baptiste Joseph de Fourier:

Francuski matematyk, fizyk i polityk. Jego największym osiągnięciem jest wykazanie i uzasadnienie możliwości reprezentacji dowolnej funkcji za pomocą nieskończonej sumy składowych harmoniczných. Swoją teorię opracował w trakcie prowadzenia działalności politycznej, jego zawierająca tę teorię ukazała się dopiero 15 lat po oddaniu do recenzji. Negatywnie ocenił ją Lagrange.

Fourier pochodził z biednej rodziny, w wieku 9 lat stracił obydwój rodziców, brak wykształcenia i wysokiego urodzenia uniemożliwiły mu naukę w szkole wyższej, sytuację zmieniła dopiero rewolucja francuska. Współpracował z Napoleonem, był prefektem dzielnicy Francji z siedzibą w Grenoble.

J.W.Cooley & J.W. Tukey:

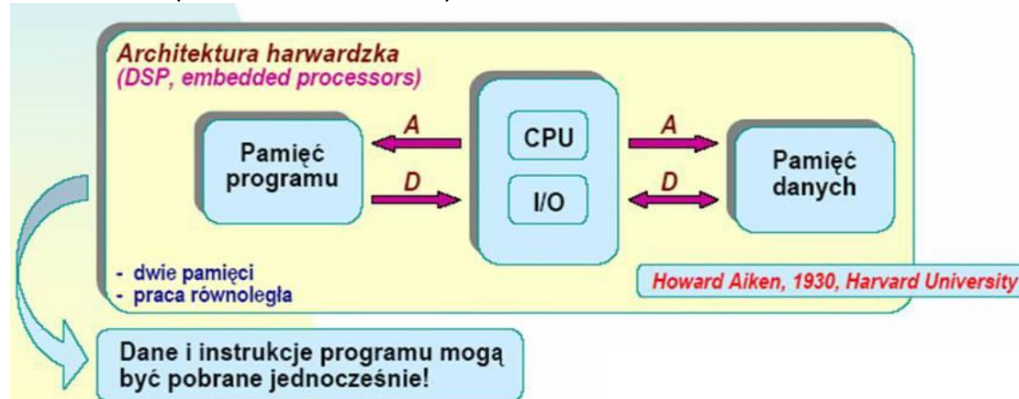
Ci panowie w 1965 opublikowali publikację „*Algorytmy dla obliczeń komputerowych zespolonych szeregów Fouriera*”, w którym proponowali udoskonalenie **DFT**, które teraz znamy jako **FFT**. Dla 1024 punktowego DFT FFT redukuje mnożenia 100 krotnie. Krok milowy w CPS, mega ważne.

Główna cecha systemów DSP:

możliwość ich pracy w trybie czasu rzeczywistego, czyli konieczność wykonania operacji obliczeniowych i sterujących w czasie wystarczającym na niezakłócony przebieg trwającego procesu

Architektura komputerowa w aspekcie DSP:

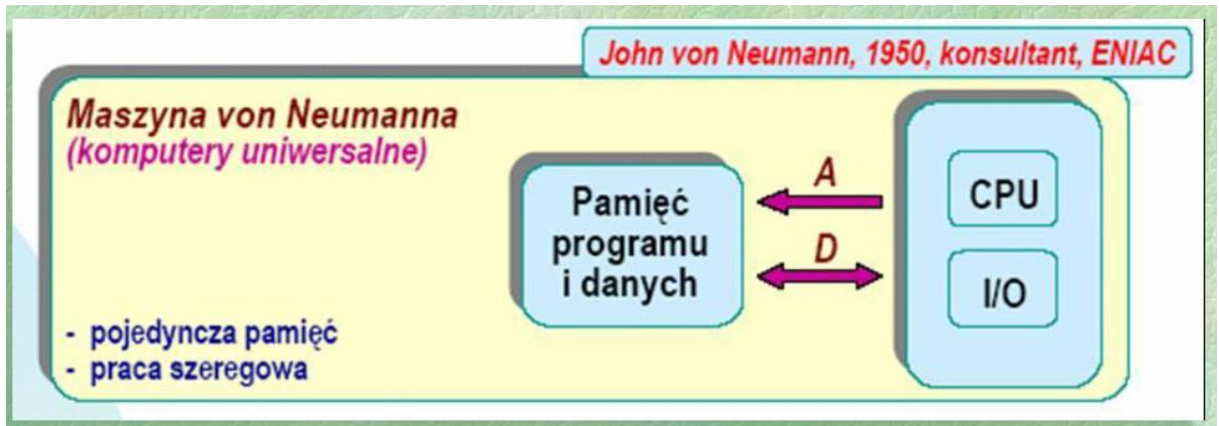
Harwardzka (autor Howard Aiken):



- Oddzielne obszary pamięci dla programu i danych, dzięki czemu dostęp do nich może być uzyskiwany jednocześnie.

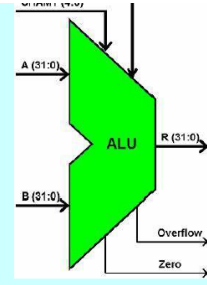
- Złożoność wynikająca z konieczności stosowania dwóch odrębnych systemów pamięciowych spowodowała, że architektura harwardzka nie zyskała popularności w rozwiązaniach uniwersalnych komputerów i w mikroprocesorach

Von Neumanna (John von Neumann)



- Brak różnicy między pamięcią programu i danymi
- Rozkazy programu mogą być dzielone na części – jedna z kodem operacyjnym, druga z adresem operandu danych
- Standard rozwoju systemów komputerowych
- Komputer ma dostęp albo do danych albo do programu (nieduży problem obecnie)

Podstawowym blokiem obliczeniowym ówczesnych komputerów była **jednostka arytmetyczno-logiczna (ALU)** oraz **rejestr przesuwany**.



- Dodawanie, odejmowanie, przesłanie – jeden cykl zegarowy
- Mnożenie, dzielenie – kilkadziesiąt cykli zegarowych

Typowe operacje DSP:

Działania elementarne:

- Pamiętanie i opóźnianie
- Mnożenie
- Dodawanie

Podstawowa operacja arytmetyczna (wykonywana w jednym cyklu zegara w procesorze):

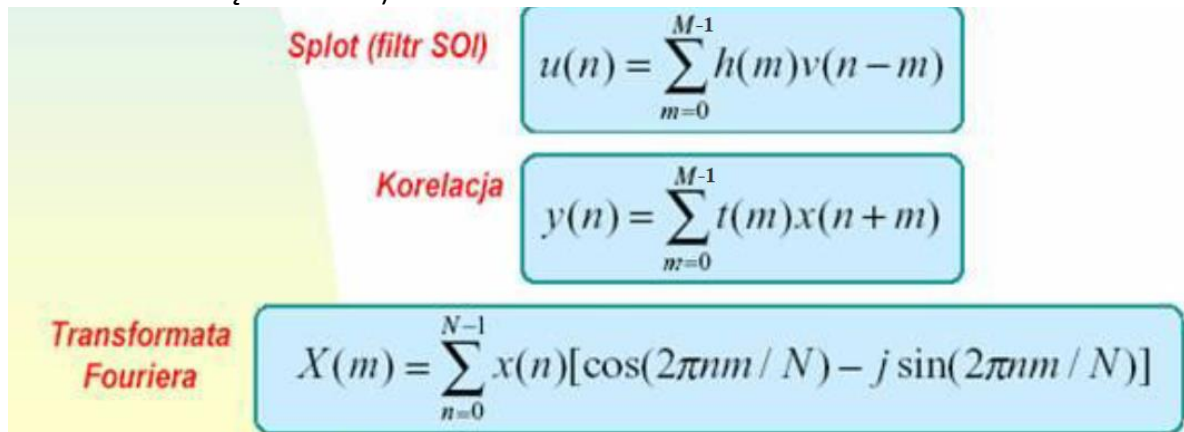
MAC – multiply and accumulate

$acc = acc + x * y$

Zadania CPS:

- Splot – podstawowa operacja filtracji sygnałów. Filtracja pozwala na zwiększenie stosunku mocy sygnału do mocy zakłóceń

- Korelacja – pozwala na porównanie sygnału z przebiegiem odniesienia (wzorcem). Redukuje wpływ składowych losowych i pomaga wykryć składowe sygnału podobne do wzorca
- Transformacja – służy do wyznaczenia widma sygnału, pozwala znaleźć charakterystyczne cechy sygnału w wybranej przestrzeni cech (np. dziedzinie częstotliwości)



Układy scalone dla DSP:

- Struktura półprzewodnikowa N-MOS pod koniec lat 70 (mogły pracować takie układy z pojedynczym zasilaniem 5V i być produkowane z zachowaniem geometrii trzymikronowej, jeden obwód – 100k tranzystorów)
- S2811, I2920, uPD7720, TMS32010

Historia rozwoju procesorów sygnałowych:

- Wczesne urządzenia DSP były na architekturze harwardzkiej
- Pierwsze mnożenie w 1 cyklu zegarowym – lata 70
- Ogromne gabarytowo urządzenie, niemożliwe komercyjne wykorzystanie
- Należało poczekać na układy scalone
- Texas Instruments – pierwszy mikroprocesor z językiem assemblerowym i symulatorem
- Dzięki symulatorom, oprogramowaniu i kompilatorom C DSP umieszczono w głównym nurcie praktyki projektowania systemów mikrokomputerowych
- Coraz większe potrzeby powodowały umieszczanie coraz większej liczby tranzystorów w urządzeniu
- Skok jakościowy z opracowaniem struktury półprzewodnikowej CMOS (geometria 0,35 mikrona)
- Zmniejszony czas propagacji i przełączania, większa gęstość upakowania tranzystorów i bramek logicznych
- 1994 – 4 miliony tranzystorów

2. Zalety cyfrowego przetwarzania sygnałów. Zastosowania CPS. Zadania CPS.

Wymagania dla systemów CPS. Wady CPS.

Zalety cyfrowego przetwarzania sygnałów:

- Programowalność i reprogramowalność
- Stabilność (np. mniejsza czułość na czynniki zewn. niż analogowe)
- Powtarzalność (te same elementy analogowe dadzą różne wyniki)
- Łatwość implementacji alg. adaptacyjnych

- Kody korekcji błędów
- Funkcje specjalne
- Kompresja danych

Zastosowania CPS:

Wszędzie xd

Urządzenia radiokomunikacyjne, telekomunikacyjne systemy satelitarne, systemy światłowodowe, osprzęt nowoczesnych central telefonicznych, urządzenia radiolokacyjne i radionawigacyjne, systemy „wizji maszynowej” i dedykowanego monitoringu, radioelektroniczna aparatura meteorologiczna i radioastronomiczna, aparatura do pomiarów geofizycznych

Zadania CPS:

- Transmisja
- Przechowywanie
- Kompresja

Wymagania dla systemów CPS:

Generalnie jest to wymagana prędkość w operacjach/sekundę oraz zakres dynamiki bitowej (stała lub zmiennoprzecinkowa)

Wykonanie operacji mnożenia i dodawania w przeciągu 1 cyklu zegarowego

Wady CPS:

- Pobierają znaczną moc z zasilacza
- Nie mogą przetwarzać sygnałów o b. dużej częstotliwości
- W środowisko analogowym wymagają przetworników AC i CA
- Bardzo trudne przetwarzanie AC i CA sygnałów słabych i silnych

3. Konwersja analogowo-cyfrowa. Harry Nyquist. Włodzimierz Kotelnikow. Claude Shannon. Próbkowanie i kwantyzacja. Prawo próbkowania Kotelnikowa-Shannona. Aliasing. Kompresja-dekompresja A-law, μ -law.

Konwersja analogowo-cyfrowa:

- Sygnały z otaczającego nas świata są sygnałami analogowymi, są one ciągłe.
- Do przetworzenia przez urządzenia CPS potrzebne jest przekształcenie takiego sygnału do dyskretnej postaci cyfrowej.

Harry Nyquist:

- Twórca twierdzenia definiującego minimalną częstotliwość próbkowania niezbędną w celu osiągnięcia precyzyjnej reprezentacji sygnału analogowego – twierdzenie Nyquista
- twórca twierdzenia o stabilności (tzw. kryterium Nyquista)
- opracował fundamenty dla prac Shannona
- Częstotliwość Nyquista to minimalna częstotliwość próbkowania, która jest co najmniej dwa razy wyższa od składowej o najwyższej częstotliwości sygnału oryginalnego w celu osiągnięcia dokładnej reprezentacji sygnału analogowego

Włodzimierz Kotelnikow:

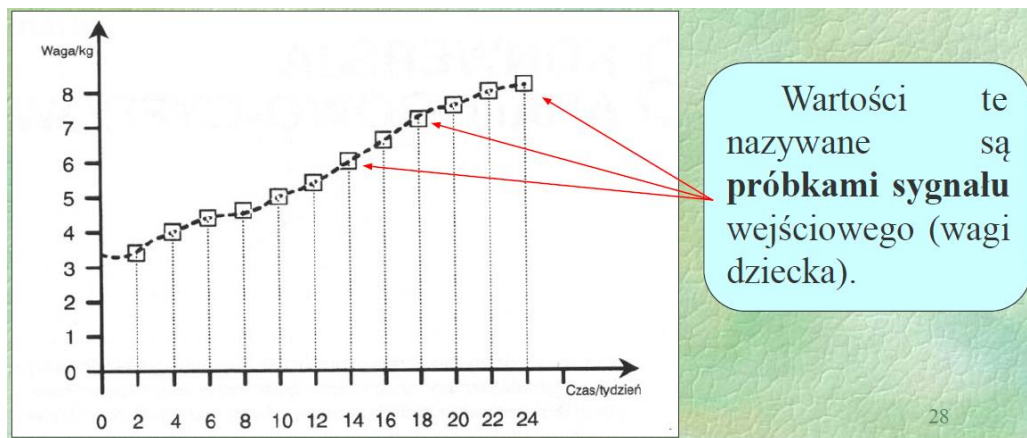
- Drugi najwybitniejszy rosyjski uczyony XX w po Aleksandrze Cariowie
- Urodzony w Kazaniu
- Twórca prawa Kotelnikowa-Shannona – twierdzenie o próbkowaniu sygnałów (sformułowane niezależnie od H. Nyquista)

Claude Shannon:

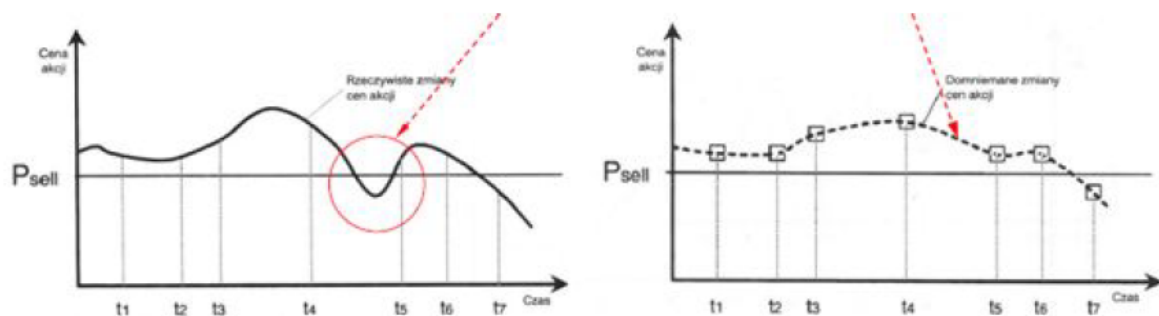
- Matematyk i inżynier amerykański
- Jeden z twórców teorii informacji, podstawowa praca: Mathematical Theory of Communication
- prowadził badania w zakresie warunków przenoszenia informacji, w 1948 opracował metodę ilościowego jej wyrażania

Próbkowanie i kwantyzacja:

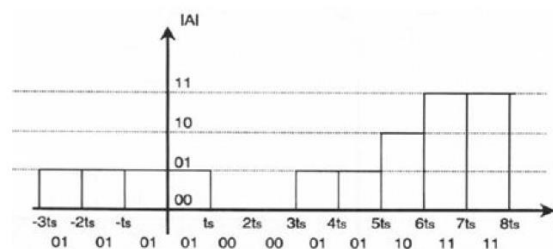
Próbkowanie – pierwszy etap procesu przekształcania postaci analogowej w cyfrową. Zamiana ciągłego sygnału na zbiór próbek sygnału wejściowego. Próbkowanie sygnału to metoda zapisu chwilowych wartości sygnału ciągłego.



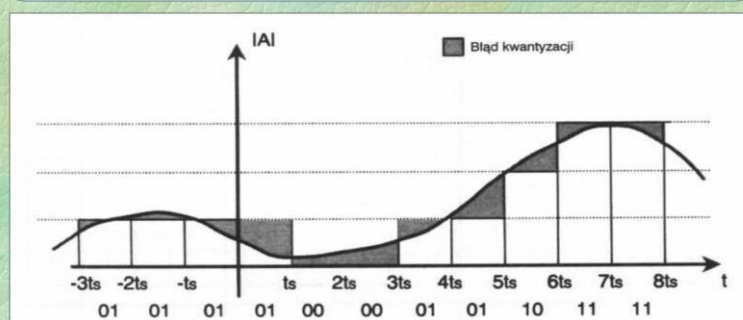
- Okres próbkowania danej wielkości jest zawsze taki sam. Krytycznym elementem mającym podstawowe znaczenie dla akwizycji pełnej informacji o zdarzeniach analogowych jest częstotliwość pobierania próbek.



- **Kwantyzacja** – sposób reprezentacji wartości chwilowych sygnału analogowego przez liczby, które mogą być wykorzystane przez urządzenia DSP.
- Inaczej – przyporządkowanie określonych odcinków amplitudowych sygnału wejściowego ustalonym przedziałom napięciowym.
- Jest realizowana przez przetwornik analogowo-cyfrowy.
- Określa się poziomy (zakresy) i wartościom sygnału przyporządkowuje się wartość zakresu, w którym się znajduje. Najczęściej na 10-12 bitach, więc kwantyzowanie z dokładnością 2^{10} lub 2^{12} .
- Różnica pomiędzy skwantyzowanymi poziomami sygnału a sygnałem oryginalnym to szum kwantyzacji.
- Kwantyzacja liniowa – poszczególne poziomy mają identyczny krok
- Nieliniowe wzorce kwantyzacyjne – np. w mowie, krok zależny od poziomu przetwarzanej amplitudy



Porównując skwantyzowane poziomy sygnału z sygnałem oryginalnym, zauważymy wprowadzone przez nas błędy:



Błędy te wywołują tzw. **szum kwantyzacji**.

👉 W zastosowaniach fonicznych efekt ten powoduje pojawienie się charakterystycznego szumu na wyjściu - stąd nazwa.

Przy zastosowaniu 10- lub 12-bitowych przetworników a/c szum ten jest na ogół pomijalny w porównaniu z innymi źródłami zakłóceń systemu.

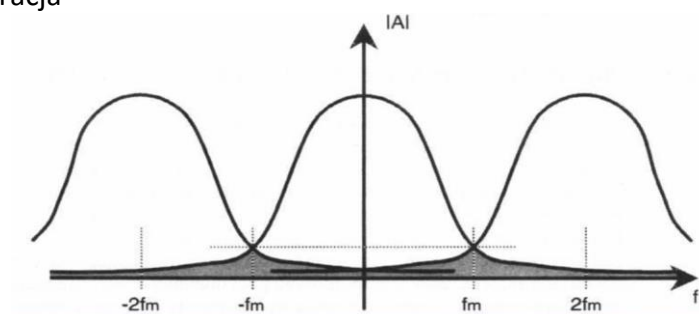
Prawo próbkowania Kotelnikowa-Shannona:

Jeśli funkcja $f(t)$ nie zawiera częstotliwości wyższych niż ω cykli na sekundę (Hz), wówczas jest ona całkowicie określona przez podanie jej rzędnych jako ciągu punktów oddzielonych od siebie o czas równy okresowi $1/2 \omega$.

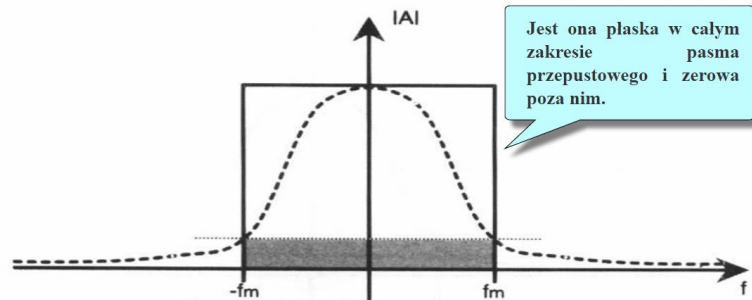
W celu osiągnięcia dokładnej reprezentacji sygnału analogowego minimalna częstotliwość próbkowania powinna być co najmniej 2 razy wyższa od składowej o najwyższej częstotliwości sygnału oryginalnego

Aliasing:

- To nakładanie się na siebie pasm częstotliwościowych powodujące interferencję sygnału wyjściowego. Dzieje się kiedy częstotliwość próbkowania f_s jest mniejsza niż dwukrotność max. Częstotliwości sygnału wejściowego f_m ($f_s < 2f_m$)
- Na skutek próbkowania sygnał wejściowy ma swój alias w przebiegu o innej częstotliwości, a próbki cyfrowe błędnie opisują sygnał, przetwarzanie będzie błędne
- Im stosunek częstotliwości (sygnału oryginalnego i przybliżonego) jest większy, tym lepsza dokładność reprezentacji cyfrowej sygnału i łatwiej odtworzyć oryginalny sygnał.
- Aby zapobiec aliasingowi należy przepuścić sygnał wejściowy przez filtr dolnoprzepustowy analogowy, który usuwa z widma wszystkie częstotliwości większe niż $f_s/2$ - częstotliwość próbkowania (antialiasing). Można by to zrobić filtrem cyfrowym, ale w praktyce jest to bardzo rzadkie – bardzo czasochłonna operacja



Idealna charakterystyka filtra antyaliasingowego przedstawiona jest na rysunku.



Źle zaprojektowany filtr może mieć tragiczne konsekwencje dla działania systemu

Kompresja-dekompresja:

- W praktyce kwantyzator ma zunifikowany rozmiar kroku, a kompresji poddawany jest sygnał wejściowy
- Dla mowy kompresji poddaje się dźwięki o wyższych amplitudach, a słabsze dźwięki są kodowane bez zmian
- Efekt finalny jest taki, jak gdyby stosowana była kwantyzacja nieliniowa
- Po przetworzeniu sygnał jest rekonstruowany na wyjściu przez dekompresję
- W Europie używa się A-law, w USA i Japonii u-law
- Techniki te zmniejszają zakres dynamiczny sygnału zwiększając jednocześnie efektywność kodowania
- Zwiększają przepustowość
- Obie techniki używane w 8 bitowych systemach telekomunikacyjnych

4. Notacja macierzowa jako uniwersalny formalizm do opisywania, konstruowania i realizacji zadań cyfrowego przetwarzania sygnałów i obrazów. Elementy rachunku macierzowego: iloczyn skalarny, suma prosta (tensorowa), iloczyn tensorowy (Kroneckera), iloczyn Hadamarda, konkatenacja. Macierze specjalne oraz ich zastosowanie.

W CPS bardzo poręcznym jest przedstawienie sygnałów jako macierzy, np. sygnału 1D jako wektora lub 2D jako macierzy

iloczyn skalarny:

$$\mathbf{A}_{N \times 1} = [a_0, a_1, \dots, a_{N-1}]^T \quad \mathbf{X}_{N \times 1} = [x_0, x_1, \dots, x_{N-1}]^T$$

$$y = \sum_{i=0}^{N-1} a_i \cdot x_i$$

$$y = [\mathbf{A}_{N \times 1}]^T \cdot \mathbf{X}_{N \times 1}$$

suma prosta (tensorowa):

$$\mathbf{A}_{N \times M} \oplus \mathbf{B}_{K \times L} = \left[\begin{array}{c|c} \mathbf{A}_{N \times M} & \\ \hline & \mathbf{B}_{K \times L} \end{array} \right],$$

$$\bigoplus_{i=0}^{N-1} \mathbf{A}_{L \times M}^{(i)} = \left[\begin{array}{c|c|c|c} \mathbf{A}_{L \times M}^{(0)} & & & \\ \hline & \mathbf{A}_{L \times M}^{(1)} & & \\ \hline & & \square & \\ \hline & & & \mathbf{A}_{L \times M}^{(N-1)} \end{array} \right]$$

iloczyn tensorowy (Kroneckera):

$$\mathbf{A}_{N \times M} \otimes \mathbf{B}_{K \times L} = \left[\begin{array}{cccc} a_{0,0} \mathbf{B}_{K \times L} & a_{0,1} \mathbf{B}_{K \times L} & \square & a_{0,N-1} \mathbf{B}_{K \times L} \\ a_{1,0} \mathbf{B}_{K \times L} & a_{1,1} \mathbf{B}_{K \times L} & \square & a_{1,N-1} \mathbf{B}_{K \times L} \\ \square & \square & \square & \square \\ a_{N-1,0} \mathbf{B}_{K \times L} & a_{N-1,1} \mathbf{B}_{K \times L} & \square & a_{N-1,N-1} \mathbf{B}_{K \times L} \end{array} \right]$$

iloczyn Hadamarda:

$$\mathbf{y} = \mathbf{A}_{N \times 1} \square \mathbf{X}_{N \times 1}$$

$$\begin{bmatrix} y_0 \\ y_1 \\ \square \\ y_{N-1} \end{bmatrix} = \begin{bmatrix} a_0 \cdot x_0 \\ a_1 \cdot x_1 \\ \square \\ a_{N-1} \cdot x_{N-1} \end{bmatrix} = \begin{bmatrix} a_0 \\ a_1 \\ \square \\ a_{N-1} \end{bmatrix} \square \begin{bmatrix} x_0 \\ x_1 \\ \square \\ x_{N-1} \end{bmatrix}$$

Dla 2 macierzy:

$$\mathbf{A}_{N \times M} = \begin{bmatrix} a_{0,0} & a_{0,1} & \square & a_{0,M-1} \\ a_{1,0} & a_{1,1} & \square & a_{1,M-1} \\ \square & \square & \square & \square \\ a_{N-1,0} & a_{N-1,1} & \square & a_{N-1,M-1} \end{bmatrix} \quad \mathbf{B}_{N \times M} = \begin{bmatrix} b_{0,0} & b_{0,1} & \square & b_{0,M-1} \\ b_{1,0} & b_{1,1} & \square & b_{1,M-1} \\ \square & \square & \square & \square \\ b_{N-1,0} & b_{N-1,1} & \square & b_{N-1,M-1} \end{bmatrix}$$

$$\mathbf{C}_{N \times M} = \begin{bmatrix} a_{0,0}b_{0,0} & a_{0,1}b_{0,1} & \square & a_{0,M-1}b_{0,M-1} \\ a_{1,0}b_{1,0} & a_{1,1}b_{1,1} & \square & a_{1,M-1}b_{1,M-1} \\ \square & \square & \square & \square \\ a_{N-1,0}a_{N-1,0} & a_{N-1,1}a_{N-1,1} & \square & a_{N-1,M-1}a_{N-1,M-1} \end{bmatrix}$$

konkatenacja:

pozioma:

$$\mathbf{A}_{N \times L} \mathbin{\square} \mathbf{B}_{N \times M} = [\mathbf{A}_{N \times L} \mid \mathbf{B}_{N \times M}]$$



$$\mathbf{C}_{M \times (N+L)} = \mathbf{A}_{M \times N} \mathbin{\square} \mathbf{B}_{M \times L} = \begin{bmatrix} a_{0,0} & a_{0,1} & \square & a_{0,N-1} & b_{0,0} & b_{0,1} & \square & b_{0,L-1} \\ a_{1,0} & a_{1,1} & \square & a_{1,N-1} & b_{1,0} & b_{1,1} & \square & b_{1,L-1} \\ \square & \square & \square & \square & \square & \square & \square & \square \\ a_{M-1,0} & a_{M-1,1} & \square & a_{M-1,N-1} & b_{M-1,0} & b_{M-1,1} & \square & b_{M-1,L-1} \end{bmatrix}$$

Pionowa:

$$\mathbf{A}_{N \times L} \mathbin{\square} \mathbf{B}_{M \times L} = \begin{bmatrix} \mathbf{A}_{N \times L} \\ \mathbf{B}_{M \times L} \end{bmatrix} \quad \mathbf{Y}_{N^2 \times 1} = \mathbf{A}_{M \times N} \mathbin{\square} \mathbf{B}_{L \times N} = \begin{bmatrix} a_{0,0} & a_{0,1} & \square & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \square & a_{1,N-1} \\ \square & \square & \square & \square \\ a_{M-1,0} & a_{M-1,1} & \square & a_{M-1,N-1} \\ \hline b_{0,0} & b_{0,1} & \square & b_{0,N-1} \\ b_{1,0} & b_{1,1} & \square & b_{1,N-1} \\ \square & \square & \square & \square \\ b_{L-1,0} & b_{L-1,1} & \square & b_{L-1,N-1} \end{bmatrix}$$

Macierze specjalne oraz ich zastosowanie:

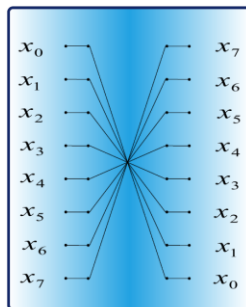


$$\mathbf{I}_8^{(2 \rightarrow)} = \begin{bmatrix} & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ 1 & & & & & & & \\ & 1 & & & & & & \end{bmatrix} \rightarrow \begin{bmatrix} x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ 1 & & & & & & & \\ & 1 & & & & & & \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix}$$

$$\mathbf{I}_8^{(\leftarrow 2)} = \begin{bmatrix} & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ 1 & & & & & & & \\ & 1 & & & & & & \end{bmatrix} \rightarrow \begin{bmatrix} x_6 \\ x_7 \\ x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ 1 & & & & & & & \\ & 1 & & & & & & \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix}$$

Macierz odbijająca

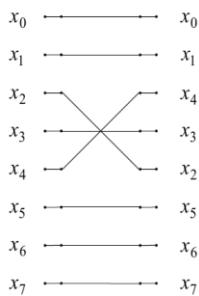
$$\mathbf{J}_N = \begin{bmatrix} & & & & & & & 1 \\ & & & & & & 1 & \\ & & & & & 1 & & \\ & & & & 1 & & & \\ & & & 1 & & & & \\ & & 1 & & & & & \\ & 1 & & & & & & \\ 1 & & & & & & & \end{bmatrix}$$



Macierz permutacji:

$$\begin{bmatrix} x_0 \\ x_1 \\ x_4 \\ x_3 \\ x_2 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} = \begin{bmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & & & & & 1 & & \\ & & 1 & & & & & \\ & & & & & & 1 & \\ & & & & & & & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix}$$

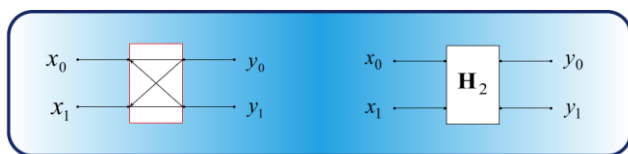
$\mathbf{P}_8^{(1)}$



Macierz Hadamarda:

$$\mathbf{H}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

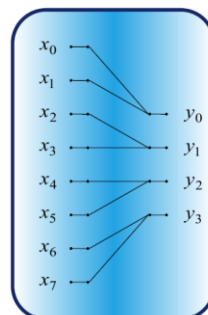
$$\begin{bmatrix} y_0 \\ y_1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \end{bmatrix}$$



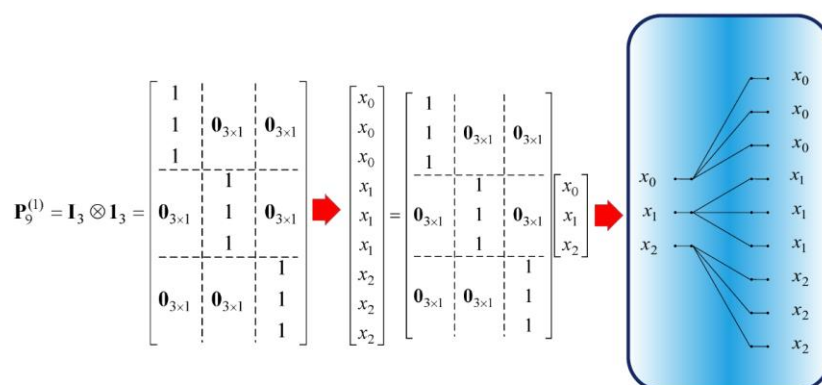
Macierz sumowania:

$$\mathbf{\Sigma}_8^{(2)} = \begin{bmatrix} 1 & 1 & & & & & & \\ & & 1 & 1 & & & & \\ & & & & 1 & 1 & & \\ & & & & & & 1 & 1 \\ & & & & & & & 1 & 1 \end{bmatrix}$$

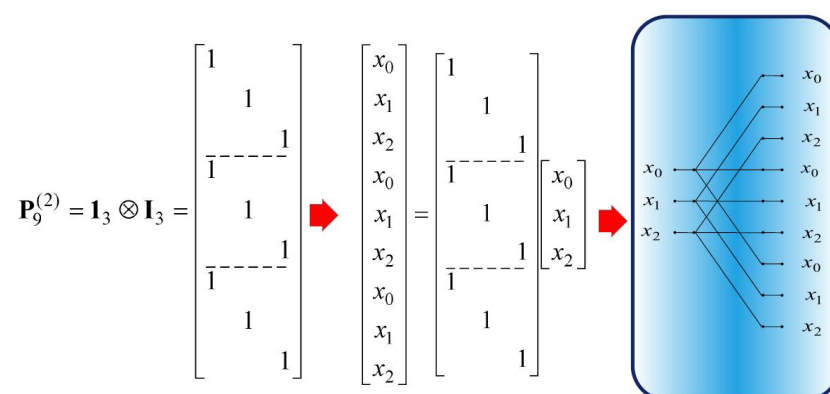
$$\begin{bmatrix} y_0 = x_0 + x_1 \\ y_1 = x_2 + x_3 \\ y_2 = x_4 + x_5 \\ y_3 = x_6 + x_7 \end{bmatrix} = \begin{bmatrix} 1 & 1 & & & & & & \\ & & 1 & 1 & & & & \\ & & & & 1 & 1 & & \\ & & & & & & 1 & 1 \\ & & & & & & & 1 & 1 \end{bmatrix} \times \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix}$$



Macierz multiplikacji



Macierz dublowania



5. Optimalizacja obliczeń wektorowo-macierzowych. Metoda Strassena. Iloczyn Winograda.

Metoda Winograda:

Shmuel Winograd – amerykański naukowiec, ur. Tel Aviv 1936.

Shmuel Winograd

$$a_0x_0 + a_1x_1 = (a_0 + x_1)(a_1 + x_0) - a_0a_1 - x_0x_1$$

$$a_0x_0 + a_1x_1 + a_2x_2 + a_3x_3 = (a_0 + x_1)(a_1 + x_0) + (a_2 + x_3)(a_3 + x_2) - a_0a_1 - a_2a_3 - x_0x_1 - x_2x_3$$

Metoda Winograda

Zależy tylko od i , więc
nie wymaga obliczeń
dla każdego j

$$y_{i,j} = \sum_{k=0}^{N-1} a_{i,k} \cdot x_{k,j} = \sum_{k=0}^{\frac{N-1}{2}} [(a_{i,2k} + x_{2k+1,j}) \cdot (a_{i,2k+1} + x_{2k,i})] - \sum_{k=0}^{\frac{N-1}{2}} a_{i,2k} \cdot a_{i,2k+1} - \sum_{k=0}^{\frac{N-1}{2}} x_{2k,j} \cdot x_{2k+1,j}$$

Zależy tylko od j , więc
nie wymaga obliczeń
dla każdego i

$$N^3 \quad \rightarrow \quad \frac{N^3}{2} + N^2$$

$$N^2 \cdot (N-1) \quad \rightarrow \quad N^2 \cdot (N-1) + N \cdot (N-1)$$

Metoda Winograda pozwala dla macierzy większych niż 4 zmniejszyć liczbę mnożeń (ok. 50% redukcji dla macierzy rzędu 128), ale zwiększa liczbę dodawań

Metoda Strassena:

METODA STRASSENÄ
DLA 2×2

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \cdot \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$$

$$\begin{aligned} M_1 &= (A_{11} + A_{22}) \cdot (B_{11} + B_{22}) \\ M_2 &= (A_{21} + A_{22}) \cdot B_{11} \\ M_3 &= A_{11} \cdot (B_{12} - B_{22}) \\ M_4 &= A_{22} \cdot (B_{21} - B_{11}) \\ M_5 &= (A_{11} + A_{12}) \cdot B_{22} \\ M_6 &= (A_{21} - A_{11}) \cdot (B_{11} - B_{12}) \\ M_7 &= (A_{12} - A_{22}) \cdot (B_{21} - B_{22}) \end{aligned}$$

$$\begin{aligned} C_{11} &= M_1 + M_4 - M_5 + M_7 \\ C_{12} &= M_3 + M_5 \\ C_{21} &= M_2 + M_4 \\ C_{22} &= M_1 - M_2 + M_3 + M_6 \end{aligned}$$

daje ona redukcję mnożeń do 7

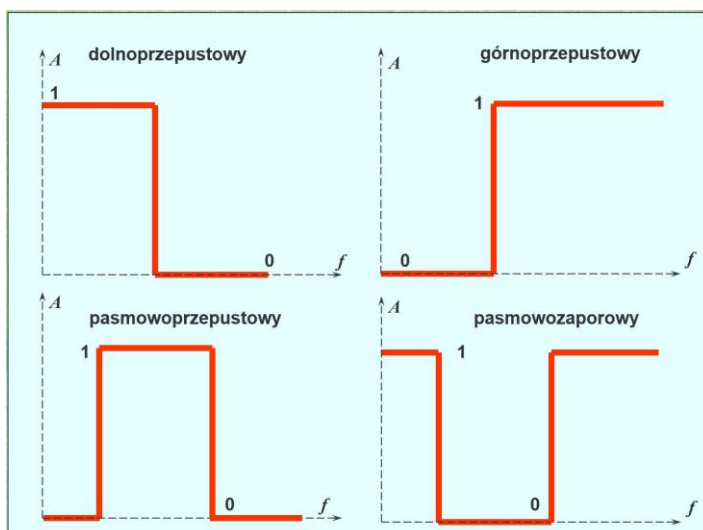
6. Metoda racjonalizacji obliczania iloczynów wektorowo-macierzowych dla macierzy posiadających specjalne struktury.

Metoda Cariowa opatentowana

- Na początek szukamy bloków macierzy zgodnych kompozycyjnie ze strukturami wzorców macierzowych
- Następnie wybieramy wzorcową procedurę i dobieramy cząstkowe procedury obliczeniowe dla znalezionych bloków
- Potem synteza końcowej wektorowo-macierzowej procedury obliczeniowej na podst. procedur cząstkowych

- Ocena efektywności algorytmu wg wybranych kryteriów
 - Często macierz wymaga wprowadzenia permutacji wierszy oraz kolumn, lub np. pomnożenia ich przez -1, aby dopasować ją do wzorca.
- Dwie dziedziny reprezentacji sygnałów. Przekształcanie sygnałów w dziedzinie częstotliwościowej, widmo. Dyskretne przekształcenie Fouriera (DFT). Dyskretne funkcje wykładnicze. Definicja prostej i odwrotnej dyskretnej transformacji Fouriera: notacja tradycyjna oraz macierzowa. Właściwości DFT.**
 - Numeryczne obliczanie widma sygnału. Szybka transformata Fouriera. Grafy sygnałowe. Operacja motylkowa. Adresacja bitowo-rewersyjna. Algorytm DFT-Winograda. Znane nankowcy z dziedziny algorytmów DFT.**
 - Uogólnienie DFT. Dyskretne bazy ortogonalne i ich własności. Szybkie algorytmy dyskretnych transformat ortogonalnych. Baza funkcji Walsha. Baza funkcji Haara. Baza funkcji Slant.**
 - Procedury obliczeniowe szybkiej transformaty w bazie dyskretnych funkcji wykładniczych. Procedury obliczeniowe szybkiej transformaty w bazie funkcji Vilenkina.**
 - Dyskretne Przekształcenie Kosinusowe (DCT). Szybkie algorytmy DCT.**
 - Filtry cyfrowe. Filtr o skończonej odpowiedzi impulsowej. Filtr o nieskończonej odpowiedzi impulsowej. Porównanie filtrów SOI i NOI. Szumy w układach filtrów.**

Filtry cyfrowe służą do redukcji szumów oraz do odtwarzania sygnałów zniekształconych.



Najprostszy typ filtru – **filtr o skończonej odpowiedzi impulsowej**, określane wzorem:

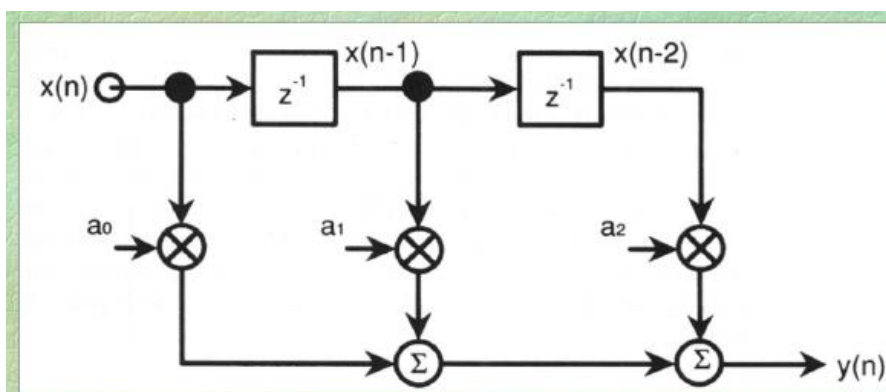
$$y_n = \sum_{-\infty}^{+\infty} a_k x_{n-k}$$

gdzie a_k – są współczynnikami filtru (wartości stałe), x_{n-k} – są danymi wejściowymi (próbki sygnału przed filtracją), y_n – dane wyjściowe (próbki sygnału po filtracji).

W teorii nieskończone współczynniki filtru, w praktyce długość ciągu jest skoczona

$$y_n = \sum_{-N}^{+N} a_k x_{n-k}$$

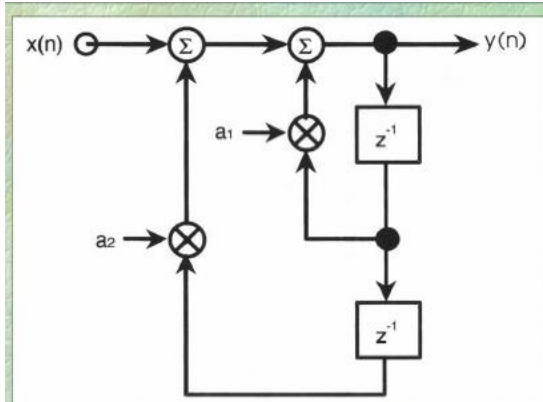
SOI – ponieważ mając skończony ciąg wejściowy elementów różnych od zera to filtr otrzyma ciąg o skończonej długości niezerowych elementów sekwencji danych wyjściowych.



NOI – filtr rekursywny, wymaga sprzężenia zwrotnego, jeśli sekwencja wejściowa stałaby się ciągiem wartości zerowych to sekwencja wyjściowa mogłaby pozostać ciągiem el. niezerowych

$$y_n = \sum_{k=0}^N a_k x_{n-k} + \sum_{k=1}^M b_k y_{n-k}$$

(w teorii również od do nieskończoności)



SOI:

- możliwość przystosowania do realizacji liniowej odpowiedzi fazowej.
- Używany tam gdzie wymagana jest liniowość charakterystyki fazowej i duży stopień kontroli odp. fazowej systemu
- Stabilny i przewidywalny

NOI:

- Charakteryzuje się kiepską odp. fazową, wyjątkowo nieliniową na brzegach pasm
- Stosując stara się uzyskać jak najbardziej strome nachylenie charakterystyki z użyciem jak najmniejszej liczby odprowadzeń, przez to słaba jakość odp. fazowej
- najczęściej stosowane przy realizacji sprzężeń międzyukładowych dla prądu zmiennego oraz w celu wygładzania(uśredniania) przebiegów.
- Ok. 5 razy efektywniejszy od soi, ale niestabilny, trzeba dobrze go projektować

Szumy w układach filtrów - wszystkie typy przetworników a/c wprowadzają błędy kwantyzacji, dodatkowo jeśli rozdzielczość wyjściowa przetwornika c/a jest niższa niż rozdzielczość DSP mogą pojawiać się dodatkowe szumy w sygnale wyjściowym.

Szumy mogą być wywołane na kilka sposobów:

- Kwantyzacja współczynników

Na przykład dokonajmy mnożenia wartości wejściowej 0,39 przez 0,4:

$$\begin{array}{r} \times 0,39 \\ 0,4 \\ \hline 0,156 \end{array}$$

Przekłamanie o jeden znak dziesiętny współczynnika może spowodować zmianę 0,4 na 0,3 i rezultat mnożenia będzie teraz inny:

$$\begin{array}{r} \times 0,39 \\ 0,3 \\ \hline 0,117 \end{array}$$

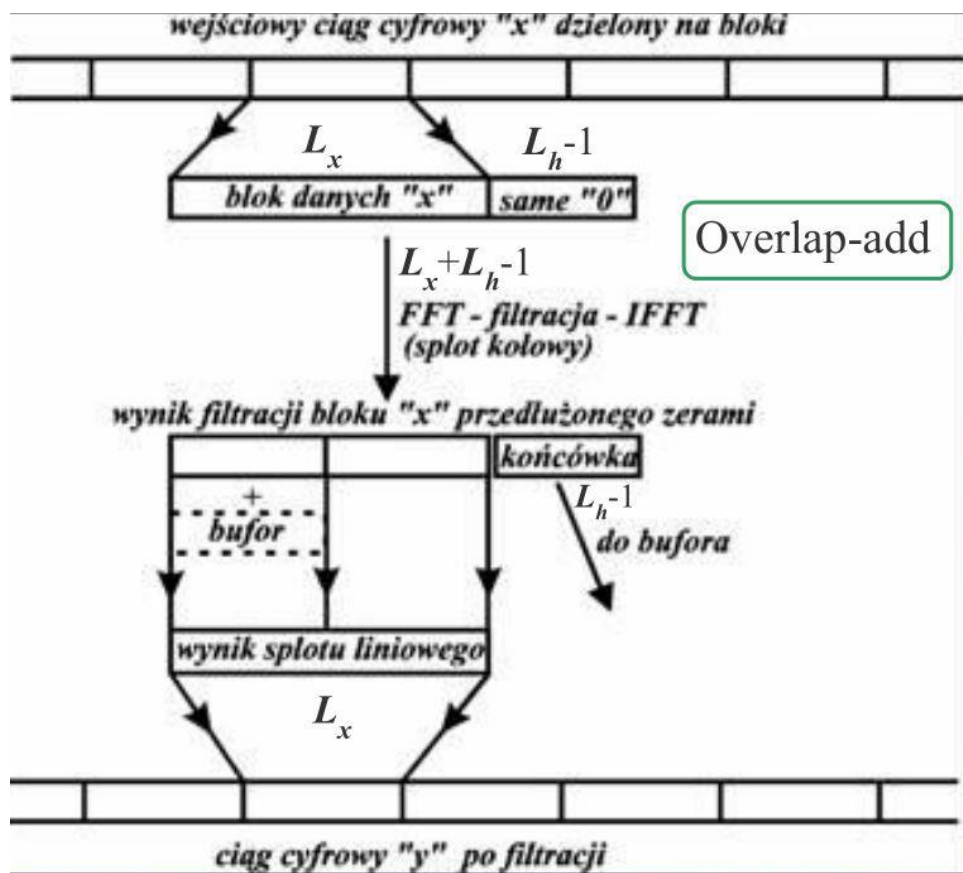
co daje błąd o wartości 25%.

- Szum obcięcia - przy wynikach wychodzących poza zakres np. dla 16 bitów wszystko wychodzące poza zakres, wpływa wyłącznie na 16 bit co daje niewielki procent, ale jednak powstaje szum.
- Overflow - za duży wynik operacji matematycznej, rezultat zostaje obcięty od góry.
- Underflow - za mały wynik operacji matematycznej, rezultat zostaje obcięty w kierunku ujemnym.
- Ograniczenia zakresu dynamicznego - ograniczenie do wielkości słowa procesora.

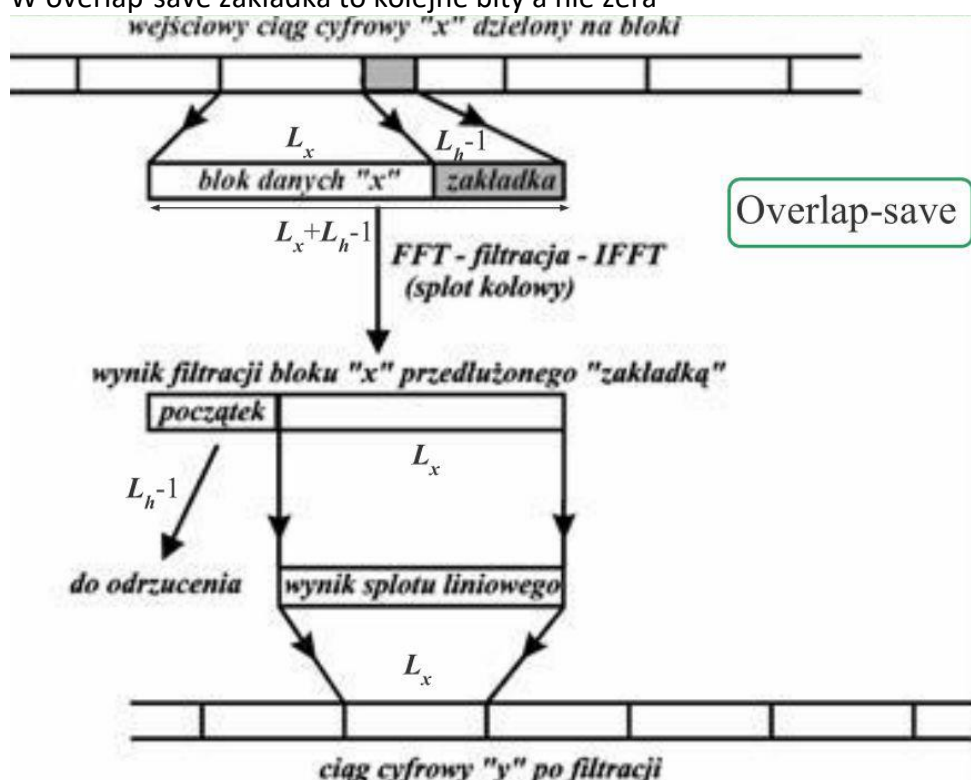
13. Splot i korelacja. Splot kołowy. Przykład wyznaczenia splotu kołowego. Korelacja kołowa. Obliczenie splotu kołowego oraz korelacji kołowej za pomocą DFT.

14. Splot liniowy. Przykład wyznaczenia splotu liniowego. Wyznaczenie splotu liniowego za pomocą splotu kołowego.

15. Realizacja splotu liniowego za pomocą splotu kołowego z wykorzystaniem FFT. Metody „overlap-add” oraz „overlap-save”. Splot dwuwymiarowy i jego zastosowanie w przetwarzaniu obrazów



W overlap-save zakładka to kolejne bity a nie zera



16. Analiza czasowo-częstotliwościowa. Krótkookresowa transformata Fouriera.

17. Technologie Falkowe. Falki. Ciągła transformata falkowa: definicja. Skala a częstotliwość. Skalowanie oraz przesunięcie.
18. Dyskretna transformata falkowa (DWT). Jednoetapowe filtrowanie: aproksymacje i współczynniki szczegółowości. Dekompozycja wielopoziomowa. Rekonstrukcja falkowa. Wieloetapowa dekompozycja i rekonstrukcja. Znanie narkowcy z dziedziny falek.
19. Procedury obliczeniowe DWT. Dyskretna transformata falkowa z wykorzystaniem falek Haara. Dyskretna transformata falkowa z wykorzystaniem filtrów Daubechies. Przykłady DWT Haara i Daubechies dla $N=8$.
20. Zrównoleglanie algorytmów CPS. Potokowość i wektoryzacja. Zrównoleglanie mnożenia wektora przez macierz. Równoległe algorytmy obliczenia spłotu. Równoległe algorytmy FFT. Równoległe algorytmy DWT.

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

$$\mathbf{Z}_{N \times 1} = [z_0, z_1, \dots, z_{N-1}]^T \quad \mathbf{Y}_{N \times M} = \|y_{i,j}\| \quad \mathbf{X}_{M \times 1} = [x_0, x_1, \dots, x_{M-1}]^T$$

$$\mathbf{Z}_{N \times 1} = \begin{bmatrix} \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square \end{bmatrix} \times \mathbf{X}_{M \times 1}$$

$\mathbf{Y}_{\frac{N}{n} \times \frac{M}{m}}^{(i,j)}$

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

$$\mathbf{Z}_{N \times 1} = [z_0, z_1, \dots, z_{N-1}]^T \quad \mathbf{Y}_{N \times M} = \|y_{i,j}\| \quad \mathbf{X}_{M \times 1} = [x_0, x_1, \dots, x_{M-1}]^T$$

The diagram shows a vertical vector $\mathbf{Z}_{N \times 1}$ on the left, followed by an equals sign, a large matrix $\mathbf{Y}_{N \times M}$ divided into four horizontal blocks, followed by a multiplication sign \times , and a vertical vector $\mathbf{X}_{M \times 1}$ on the right. The central block of the matrix is labeled $\mathbf{Y}_{N \times M}^{(i)}_n$. Above the matrix, the definition $\mathbf{Y}_{N \times M} = \|y_{i,j}\|$ is given. The vectors are defined as $\mathbf{Z}_{N \times 1} = [z_0, z_1, \dots, z_{N-1}]^T$ and $\mathbf{X}_{M \times 1} = [x_0, x_1, \dots, x_{M-1}]^T$.

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

$$\mathbf{Z}_{N \times 1} = [z_0, z_1, \dots, z_{N-1}]^T \quad \mathbf{Y}_{N \times M} = \|y_{i,j}\| \quad \mathbf{X}_{M \times 1} = [x_0, x_1, \dots, x_{M-1}]^T$$

The diagram shows a vertical vector $\mathbf{Z}_{N \times 1}$ on the left, followed by an equals sign, a large matrix $\mathbf{Y}_{N \times M}$ divided into four vertical blocks, followed by a multiplication sign \times , and a vertical vector $\mathbf{X}_{M \times 1}$ on the right. The central block of the matrix is labeled $\mathbf{Y}_{N \times M}^{(j)}_m$. Above the matrix, the definition $\mathbf{Y}_{N \times M} = \|y_{i,j}\|$ is given. The vectors are defined as $\mathbf{Z}_{N \times 1} = [z_0, z_1, \dots, z_{N-1}]^T$ and $\mathbf{X}_{M \times 1} = [x_0, x_1, \dots, x_{M-1}]^T$.

**21. Procesory sygnałowe. Realizacja zadania DSP. Najpopularniejsze zastosowania.
Generacje procesorów DSP. Charakterystyka procesorów DSP. Cechy procesorów
DSP. Typy procesorów DSP.**

**22. Przyspieszenie obliczeń: walka i jedność przeciwności. Racjonalizacja obliczeń a
prawo zachowania.**