

Politechnika  Białostocka

Wydział Elektryczny

Katedra Telekomunikacji i Aparatury Elektronicznej

Instrukcja do zajęć laboratoryjnych z przedmiotu:

Przetwarzanie Sygnałów

Kod: TS1C400027

Temat ćwiczenia:

**Badanie podstawowych charakterystyk i właściwości filtrów
o skończonej odpowiedzi impulsowej**

Opracował: dr inż. Dariusz Jańczak

Białystok 2018

**Temat: Badanie podstawowych charakterystyk i właściwości filtrów
o skończonej odpowiedzi impulsowej**

1. Cel ćwiczenia

Celem ćwiczenia jest poszerzenie wiedzy studentów oraz nabycie przez nich umiejętności z zakresu syntezy i analizy filtrów cyfrowych o skończonej odpowiedzi impulsowej (SOI). W ramach zajęć studenci ugruntowują umiejętność sformułowania specyfikacji filtrów cyfrowych oraz nabywają podstawowe umiejętności z zakresu syntezy filtrów SOI metodą okien czasowych oraz jej weryfikacji przy wykorzystaniu narzędzi komputerowego wspomagania projektowania. Studenci zdobywają umiejętność realizacji sprzętowej filtrów cyfrowych oraz zapoznają się z praktycznymi aspektami ich implementacji. Studenci utrwalają umiejętność wykonania pomiarów ich charakterystyk. Prowadząc porównanie właściwości filtrów SOI z filtrami o nieskończonej odpowiedzi impulsowej, studenci doskonalą umiejętność prowadzenia analizy porównawczej. Studenci doskonalą również umiejętność opracowania dokumentacji dotyczącej realizacji zadania inżynierskiego.

2. Zagadnienie do opracowania przed przystąpieniem do zajęć

2.1 Przed przystąpieniem do zajęć należy opracować (na podstawie materiałów z wykładu "Przetwarzanie Sygnałów 1") następujące zagadnienia teoretyczne:

- podstawowe właściwości filtrów cyfrowych typu SOI: struktura, opis transmitancyjny, odpowiedzi czasowe, charakterystyki częstotliwościowe;
- projektowanie dolnoprzepustowych filtrów SOI metodą okien czasowych;
- metody projektowania środkowo i górnoprzepustowych filtrów SOI metodą okien czasowych.

3. Program ćwiczeń

3.1. Wykorzystując podane funkcje programu Matlab przeprowadzić obserwacje i porównanie kształtu okien czasowych: prostokątnego (boxcar.m), trójkątnego (triang.m), bartletta (bartlett.m), blackmana (blackman.m), hamminga (hamming.m), hanninga (hanning.m).

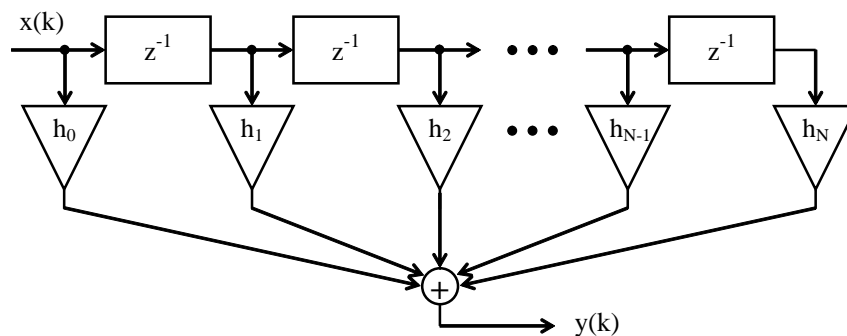
- 3.2. Zaprojektować dolnoprzepustowy filtr SOI o zadanych przez prowadzącego parametrach.
- 3.3. Zbadać wpływ długości okna na charakterystyki czasowe i częstotliwościowe filtra.
- 3.4. Zbadać wpływ różnych kształtów okien na charakterystyki czasowe i częstotliwościowe filtra.
- 3.5. Korzystając z filtra otrzymanego w punkcie 3.2 zaprojektować środkowoprzepustowy filtr o częstotliwości środkowej podanej przez prowadzącego. Należy wykorzystać metodę przekształcenia dolnoprzepustowego filtra SOI.
- 3.6. Zaprojektować filtr SOI na podstawie charakterystyki czasowej dolnoprzepustowego filtra NOI. Przy projektowaniu skorzystać z funkcji fir2.m. Zbadać zbieżność charakterystyk częstotliwościowych filtrów w zależności od rzędu filtra SOI.
- 3.7. Zaimplementować na procesorze DSP wybrane przez prowadzącego filtry SOI. Zdjąć ich charakterystyki amplitudowe. Porównać otrzymane charakterystyki amplitudowe z charakterystykami wyznaczonymi symulacyjnie.

4. Realizacja ćwiczeń

Filtry SOI należy zrealizować w formie filtrów transwersalnych opisanych transmitancją:

$$H(z) = h_0 + h_1z^{-1} + h_2z^{-2} + \dots + h_Nz^{-N}$$

gdzie h_i - stałe współczynniki.



Rys. 1. Struktura filtra transwersalnego rzędu N

Badania symulacyjne należy przeprowadzić w środowisku Matlab.

Wymagane charakterystyki można uzyskać przy pomocy następujących funkcji:

- `freqz(L,M,Pf,Fs)` - charakterystyki częstotliwościowe filtra cyfrowego

gdzie L, M - licznik, mianownik transmitancji

Pf - wektor punktów w których liczona jest odpowiedź

Fs - częstotliwość próbkowania w Hz

np.: `freqz([1], [1, -0.2], 100:10:1000, 5000)`

- `dimpulse(L,M,to)` - odpowiedź impulsowa dyskretnego układu liniowego

- `dstep(L,M,to)` - odpowiedź skokowa dyskretnego układu liniowego

- `conv(L1,L2)` - spłot, mnożenie wielomianów

np. `conv([1, -1], [1, -1])`

- `[hn]=fir2(N,F,M,okno(N+1))` - projektowanie filtra SOI metodą okien czasowych

gdzie: hn - współczynniki filtra SOI

N - rząd filtra SOI

F, M - opis idealnej wymaganej charakterystyki częstotliwościowej

(F - wektor częstotliwości unormowanych względem $f_s/2$,

M - wartości charakterystyki amplitudowej dla częstotliwości F)

okno - rodzaj funkcji okna czasowego jak w p.3.1. np. `bartlett(N+1)`

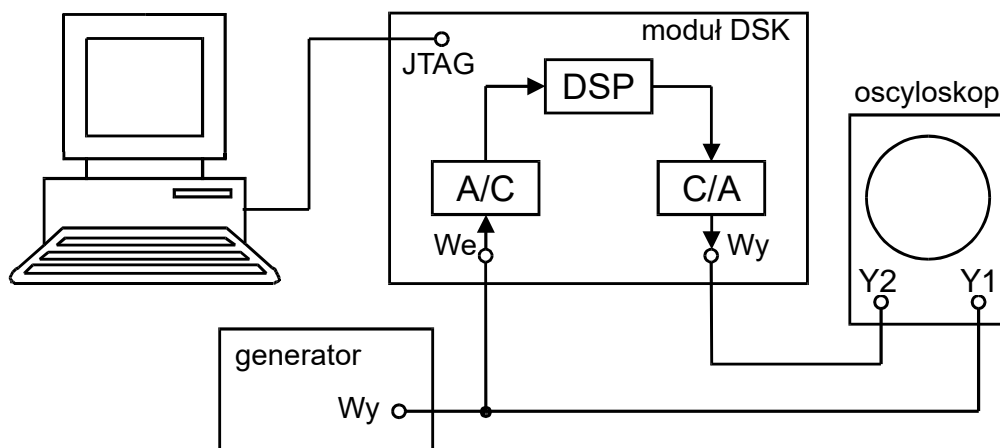
- funkcje okien wymienione w p. 3.1.

- `round(x)` - zaokrąglenie wartości liczby x do najbliższej całkowitej

- `ifft(X)` - odwrotna transformata Fouriera

- `real(z)` - część rzeczywista liczby z

Badania eksperymentalne należy przeprowadzić w układzie połączonym według schematu przedstawionego na Rys. 2.

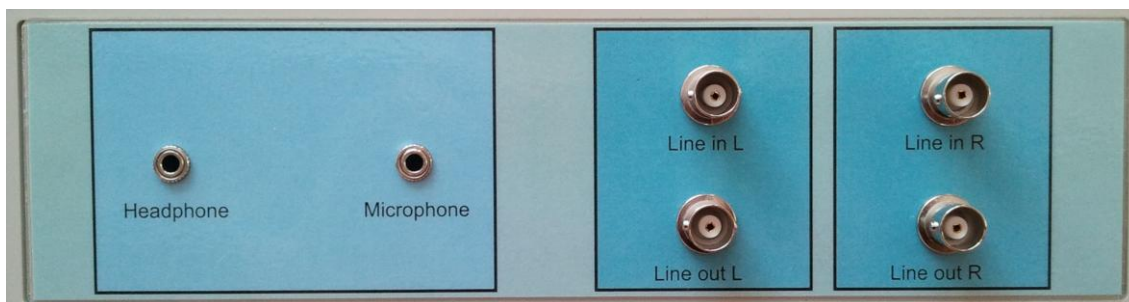


Rys. 2. Schemat połączeń stanowiska laboratoryjnego DSP

Uwaga!

Przed dołączeniem generatora do układu DSK upewnić się, że amplituda sygnału nie przekracza 1V.

W ćwiczeniu należy wykorzystać kanał lewy (patrz rys. 3 : "Line in L" - wejście, "Line out L" - wyjście).



Rys. 3. Widok płyty czołowej modułu

Filtr cyfrowy SOI w postaci filtra transversalnego realizowany jest przy wykorzystaniu procesora DSP, którego oprogramowanie zostało przygotowane jako projekt *PS_lab6* (środowisko uruchomieniowe: Code Composer Studio). Po połączeniu układu pomiarowego, należy wygenerować plik *dane.dat* zawierający współczynniki filtra realizowanego na DSP. Można to zrealizować wykorzystując program *filtr_SOI.m* uruchamiany w środowisku Matlab. Program *filtr_SOI.m* znajduje się w katalogu *D:\DSP_C6x* i wywołany jest w następujący sposób:

$$\text{filtr_SOI}(L)$$

gdzie L - współczynniki transmitancji filtra transversalnego (elementy odpowiedzi impulsowej).

Program *filtr_SOI.m* automatycznie generuje plik konfiguracyjny *dane.dat* i zapisuje go w odpowiednim katalogu projektu DSP.

Częstotliwość próbkowania ustawiona jest na $f_s = 48 \text{ kHz}$.

Kolejnym krokiem jest uruchomienie środowiska Code Composer Studio i otwarcie projektu *PS_lab6* znajdującego się w katalogu *D:\DSP_C6x\PS_lab6*. lub jeśli projekt ten jest widoczny w oknie *Project Explorer* to wystarczy ustawić jego status jako aktywny (*Active*). Projekt należy poddać kompilacji i linkowaniu (*Build / Ctrl+B*), po czym należy załadować program do modułu DSP (*Debug / F11*) (jednocześnie nastąpi przejście do trybu uruchamiania *Debug*). Jeśli nie zostały zgłoszone błędy można uruchomić program (polecenie *Resume/Run* lub *F8*). Każdorazowa zmiana parametrów

filtrów w pliku *dane.dat* wymaga powtórzenia poleceń: *Build* oraz *Resume/Run*.

Odpowiedzi czasowe filtrów należy zaobserwować na oscyloskopie, po czym można je przenieść do komputera PC i zapisać w pliku korzystając z programu *freecapture* lub *FreeView*.

5. Sprawozdanie powinno zawierać:

- procedurę projektową filtrów SOI,
- odpowiedzi czasowe,
- charakterystyki amplitudowe filtrów,
- analizę wpływu szerokości i rodzajów okien na charakterystyki filtrów,
- porównanie z filtrami NOI,
- uwagi i wnioski nasuwające się w trakcie wykonywania ćwiczenia.

6. Wymagania BHP

W trakcie realizacji programu ćwiczenia należy przestrzegać zasad omówionych we wstępie do ćwiczeń, zawartych w: „Regulaminie porządkowym w laboratorium” oraz w „Instrukcji obsługi urządzeń elektronicznych znajdujących się w laboratorium z uwzględnieniem przepisów BHP”. Regulamin i instrukcja są dostępne w pomieszczeniu laboratoryjnym w widocznym miejscu.

7. Literatura:

1. Lyons R., *Wprowadzenie do cyfrowego przetwarzania sygnałów*, WKŁ, Warszawa, 2010.
2. Zieliński T., *Cyfrowe przetwarzanie sygnałów: od teorii do zastosowań*, WKŁ, Warszawa, 2009.
3. Smith S. W., *Cyfrowe przetwarzanie sygnałów: praktyczny poradnik dla inżynierów i naukowców*, Wydawnictwo BTC, Warszawa, 2007.
4. Stranneby D., *Cyfrowe przetwarzanie sygnałów: metody, algorytmy, zastosowania*, BTC, Warszawa, 2004