DSP LAB 4

BY EE23BTECH11044 - Nimal Sreekumar

1) In MATLAB

1. Enter filterDesigner in MATLAB command window
2. It opens a filter designer window.
3. Select the following options:
a. bandpass
b. FilterOrder = Minimum
c. Density factor = 20
d. Frequency spec : Sampling freq = 48000, fstop1 = 500, fpass1 = 1500, fpass2 = 8000 ; fstop2 = 9000

4. Press design filter.
5. You can see filter response plot with band-pass characteristics
6. Go to File-¿Export and export MATLAB file. This will store all filter coefficients.

7. Write a MATLAB code to
a. read the above values into a variable by using load command. Consider 20 filter coefficients
b. convert them into fixed point Q(2,14). Store these values into a file.
c. Generate 4 sinewaves of 100Hz, 2000Hz, 6000Hz, 11000Hz and sample at 48000Hz. consider 5 cycles
d. Convert these values into Q(2,14)format and store them into file(s).
8. Write MATLAB code to get the output for these 4 signals using the designed filter (consider 20 filter coefficients). Plot outputs for 4 signals

In Verilog

1. read filter coefficients file and 4 signal files
2. write modules to find output values for each of the signal, when the given filter is applied
3. Store Output results into different files.

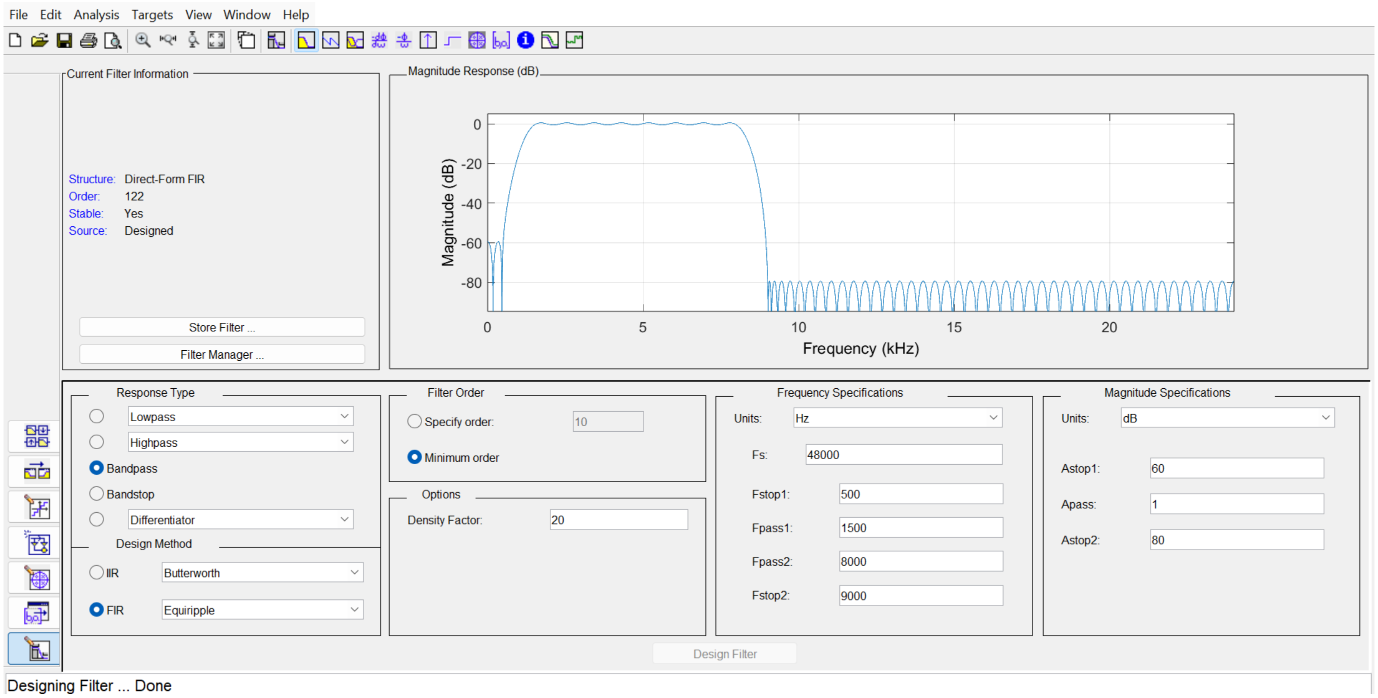
4. Read the output values in MATLAB and make plots and verify that output plots match with those from MATLAB.

You need to submit both codes and writeup on FIR filter design including relevant maths

ANSWER :

In the filter design section:

We have designed a bandpass filter with pass range of 1500Hz to 8000Hz. From the design



we can see that it is a filter with order 122 I.e, there are 123 coefficents for the filter. We have also just used a FIR filter for easy analysis.

FIR filter eqaution:

$$y[n] = \Sigma_{k=0}{}^{N} h[k]x[n-k]$$

where $h[k]$ is the filter coefficents and $x[n]$ is the input signal.

Band-Pass :

A bandpass filter is designed by combining a low-pass and high-pass filter

The ideal bandpass filter response in the frequency domain:

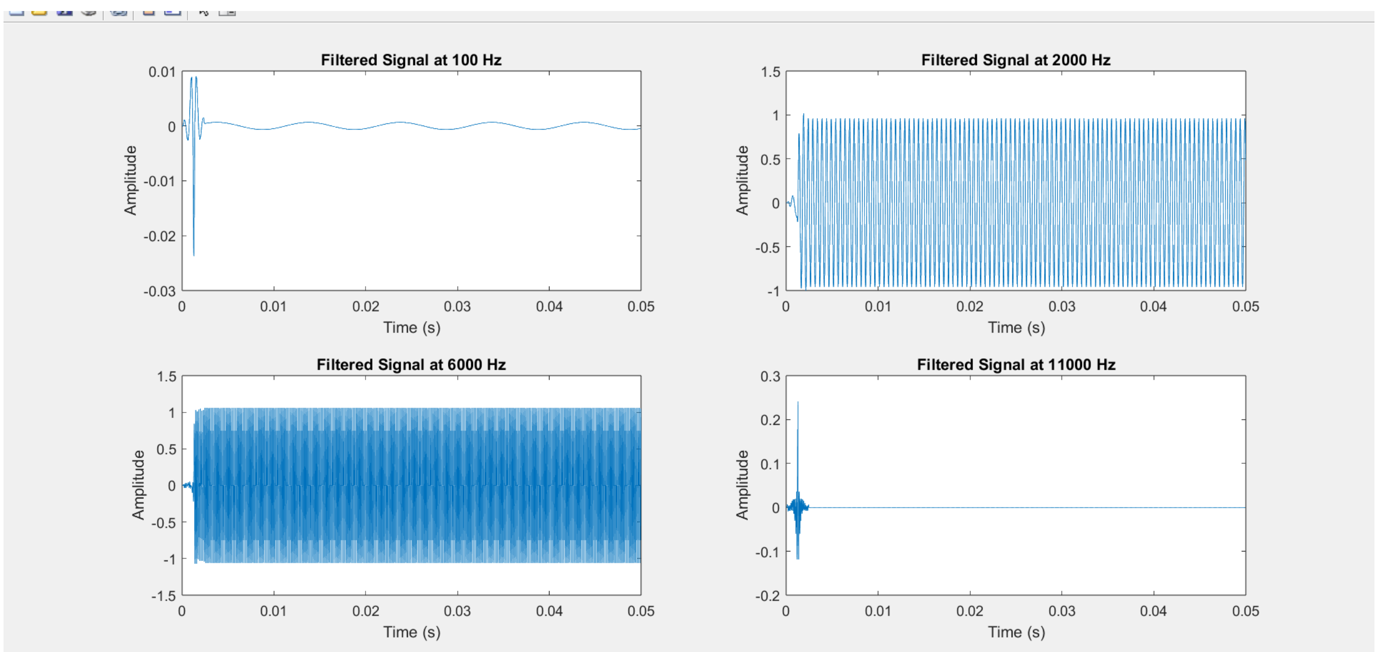
$$H(f) = 1, for f_{pass1} < f < f_{pass2}$$

and 0 in all other cases.

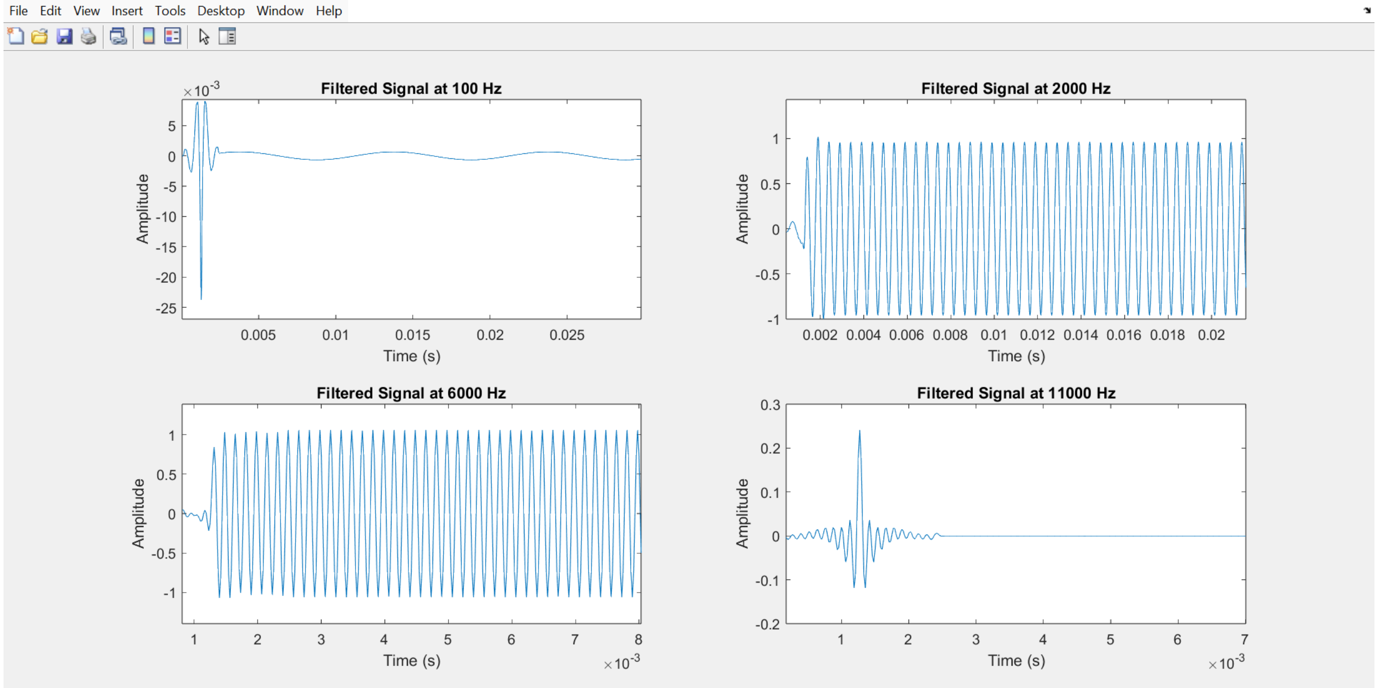
```
clc; clear; close all;

%% Load filter coefficients
load('DSPLAB.mat');

% Q(2,14)
wl = 16;
fl = 14;
b_fixed = fi(Num, true, wl, fl);
fileID = fopen('filter_coeff_fixed.txt', 'w');
fprintf(fileID, '%d\n', b_fixed.int);
fclose(fileID);

% Generate sine waves
fs = 48000;
frequencies = [100, 2000, 6000, 11000];
filter_length = length(Num);

total_samples = max(5 * fs ./ frequencies, filter_length);
max_samples = max(total_samples);
t = (0:1/fs:(max_samples-1)/fs)';
signals = sin(2 * pi * frequencies .* t);

% Convert signals to Fixed Point Q(2,14)
signals_fixed = fi(signals, true, wl, fl);
fileID = fopen('signals_fixed.txt', 'w');
fprintf(fileID, '%d %d %d %d\n', signals_fixed.int);
fclose(fileID);

% Apply FIR filter
filtered_signals = filter(Num, 1, signals);
```

```
% Plot results
figure;
for i = 1:length(frequencies)
    subplot(2,2,i);
    plot(t, filtered_signals(:,i));
   title(['Filtered Signal at ',
   num2str(frequencies(i)), ' Hz']);
    xlabel('Time (s)');
    ylabel('Amplitude');
end
```

The following are the results. From the results we can easily see that the sine waves with frequencies below 1500Hz (which is 500Hz) and above 8000Hz (which is 11000Hz) are not been allowed to pass. Only frequencies between 1500Hz and 8000Hz , which are 2000Hz and 6000Hz have been passed. Essentially showing the correct characteristics for a band-pass filter.



zoomed:


File Edit View Insert Tools Desktop Window Help
Filtered Signal at 100 Hz
Filtered Signal at 2000 Hz
Filtered Signal at 6000 Hz
Filtered Signal at 11000 Hz
Amplitude
Time (s)
×10^-3