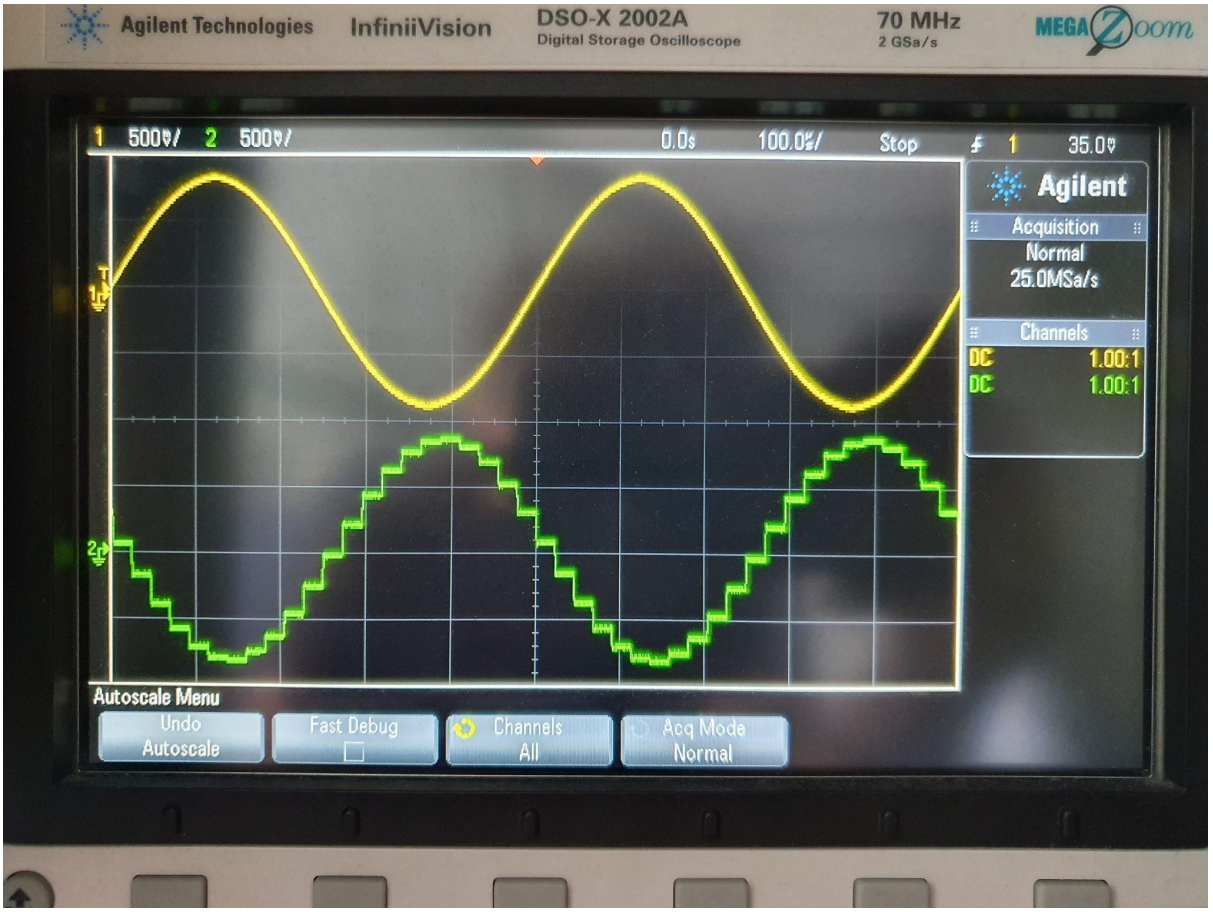
Date: 23.05.2022 8:15 - 12:00
Author: Sławomir Batruch

EDISP - Lab 5
Hardware implementations

Task 1

Running a trivial filter on the FPGA

After setting the connections per the lab instruction and running the required software, the obtained signal is shown below. The device passes the signal without changing it, apart from the phase shift. Frequency from the frequency gen is 2 kHz, wave is sinusoidal. Green sinusoidal wave is from the MyRIO, yellow is from the function generator.



We can see that the signals are out of phase. For the yellow signal, -1 of full scale appears at roughly the same time as 1 of full scale for the green signal. Signals are pi radians out of phase, which means that myRIO is inverting the signal.

In the next point, I switched into a rectangular wave on the frequency generator and zoomed in on the oscilloscope to better measure the delay between signals. I measure the time delay between ground reference of one signal and ground reference of the other signal during the transition.

From the picture we see that the difference is a bit more than 3 full divisions, which is roughly 7.4 us * 3.25 = 24.05 us. This duration is slightly longer than the duration of 1 sample in the chosen 44.1 kHz sampling frequency. The time of one sample is 1/44.1 kHz which is equal to roughly 22.7 us

In the next point, filter coefficients were changed to [0,1,0]. The delay was measured using the same way as in the previous point.

And we see that in this case, the delay is about 2 times bigger than in the previous point. It is 23 us * 2 = 46 us. This means that changing the coefficients to [0,1,0] has resulted in 2 sample delay.

As we reach the maximum frequency for our nyquist range, the signal from the FPGA starts to become unreadable and ‘noisy’. The signal from the function generator is shown properly on the oscilloscope. On the frequency spectrum in the RT_main window, after reaching fs/2, the frequency spectrum spike “bounces” from the maximum value and starts going back (it enters a different nyquist range).

The staircase effect present on the signal from myRIO is caused by the filter and ADC accuracy and ability to process the signal.

Task 3, 4, 5, 6

Design, implementation and testing of 6th order FIR filter on FPGA

Regarding the filter in the block diagram, I copied proper parts multiple times and connected them in the same manner as previous filter coefficients.

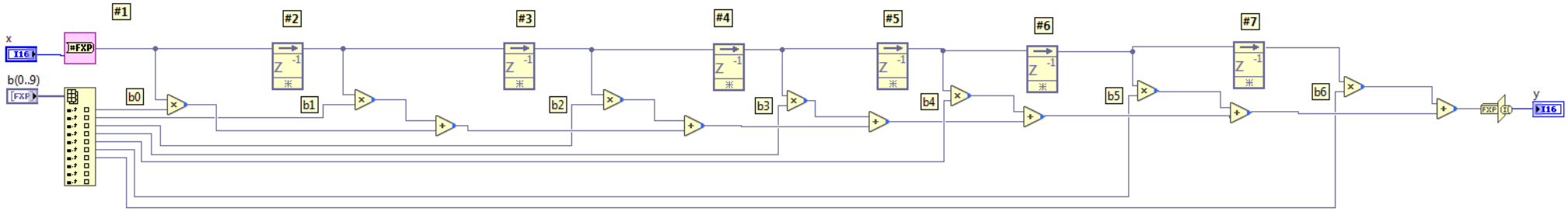
Then, I launched MATLAB to design a 6th order FIR filter. I chose to design lowpass filter. It will pass frequencies from 0 to 5 kHz, and will start to attenuate frequencies higher than 5 kHz.

After that I copied the filter coefficients from MATLAB to the RT_main window and restarted myRIO.

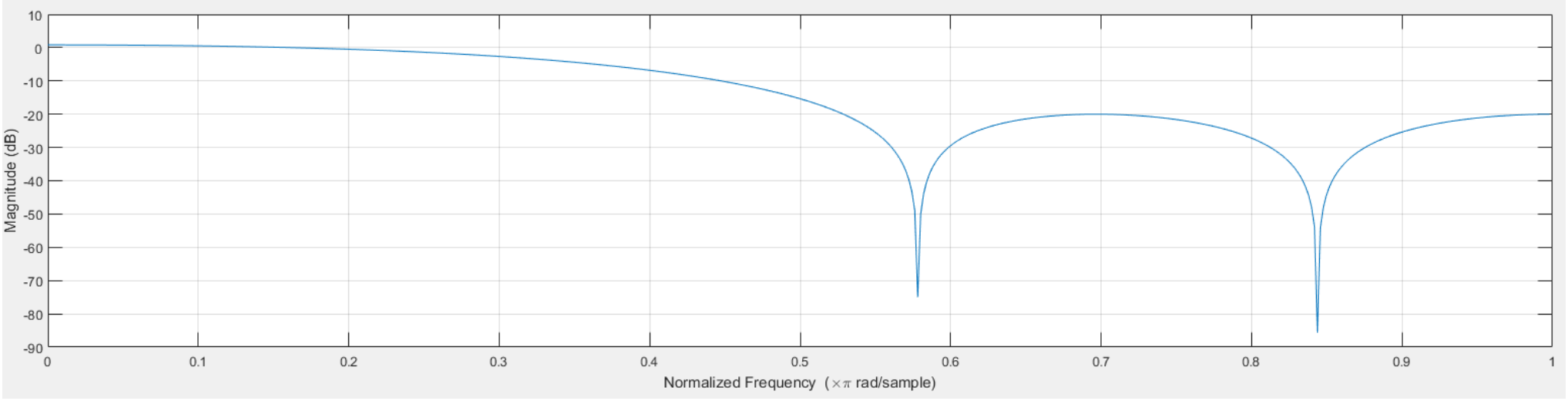
The testing consisted of sweeping the frequency on the function generator and observing the amplitude of the green signal on the oscilloscope depending on the current frequency on the frequency generator.

It was noticeable that at 5 kHz the amplitude of the signal started to drop, reaching almost 0 amplitude at about 12 kHz. These values check out with the magnitude response provided below: 0.58 * (fs/2) = 12.789 kHz is the first minimum (about -75 dB) of the magnitude response. If we increase the frequency further, some very small amplitude signal might appear due to response being -20 dB at 0.7 * (fs/2) = 15.435 kHz.

Below I show figures showing the FIR filter block diagram, as well as the magnitude response of the designed filter.



Magnitude response of the designed filter:



MATLAB code - Appendix

MATLAB code used for filter design in task 3,4,5,6:

```
N = 6; % Order of filter
Fp = 5e03; % passband-edge frequency 5 kHz
Fs = 44100; % 44.1kHz sampling freq

Fstop = Fp/(Fs/2);
eqnum = firceqrip(N, Fstop, [0.1, 0.1], 'passedge');
freqz(eqnum);
```