**ECE 471**

**Digital Signal Processing**

**FPGA Project**

**Swapnil Acharya**

**4/26/2019**

**Objective:**   
The objective of this project is to implement a Digital Signal Processing (DSP) System on a Nexys4 board using on board ADC, DSP and anti-imaging filter.

**Design:**

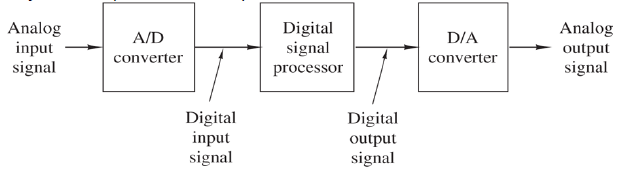
  
[1] Figure 1: System Design Block Diagram

Figure 1 gives a general overview of the system design. This design will take analog Sine wave with variable frequency, amplitude of 1Volt (V) peak to peak (pp) and an offset of 500mV as input. XADC in the nexys 4 board will used to read this analog input. The input will be digitized using the onboard XADC. The XADC sampling rate (conversion rate) is set at 100kHz. The digitized samples will then be filtered using the FIR compiler IP in Vivado which uses onboard DSP chips. The filtered signal will be changed to PWM signal where the width of the signal will represent voltage levels. The PWM signal will then be filtered out using the on-board Sallen-Key Butterworth Low-pass 4th Order Filter as the anti-imaging filter.

**Procedure:**

1. A new Project was created in Vivado targeting the Nexys 4 board.
2. From the IP catalog XADC IP was instantiated with the following settings

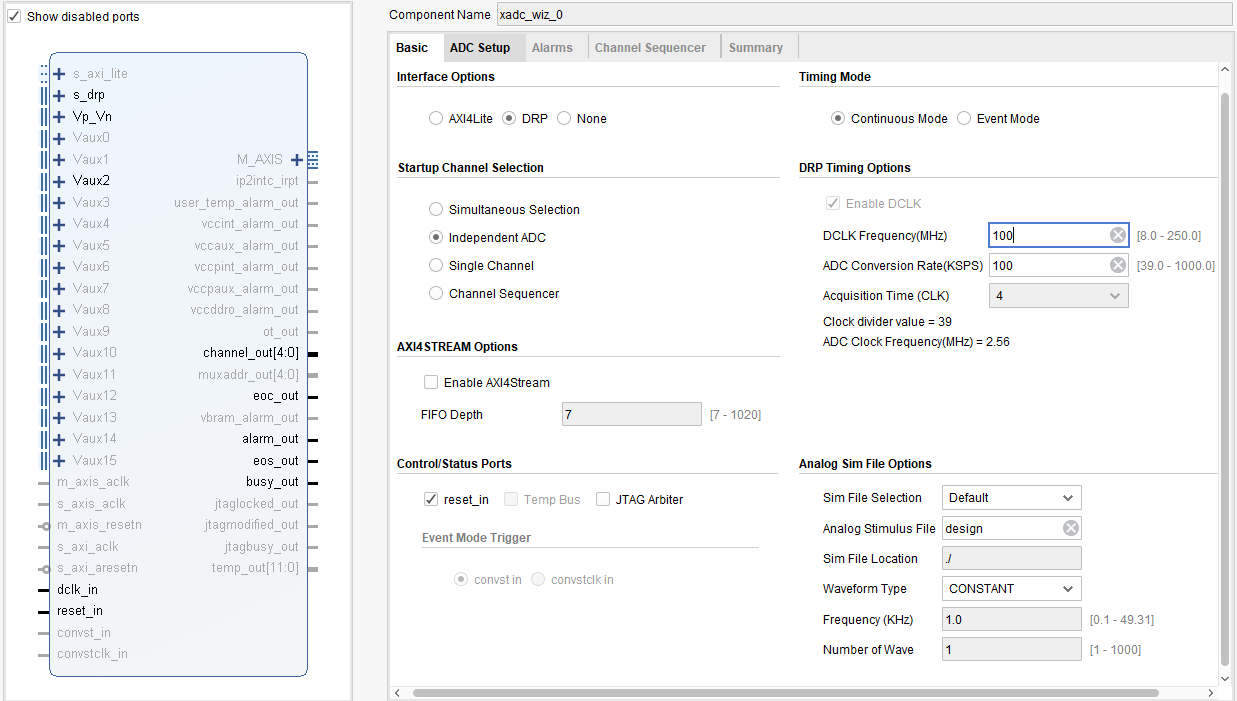


Figure 2a: XADC IP settings1

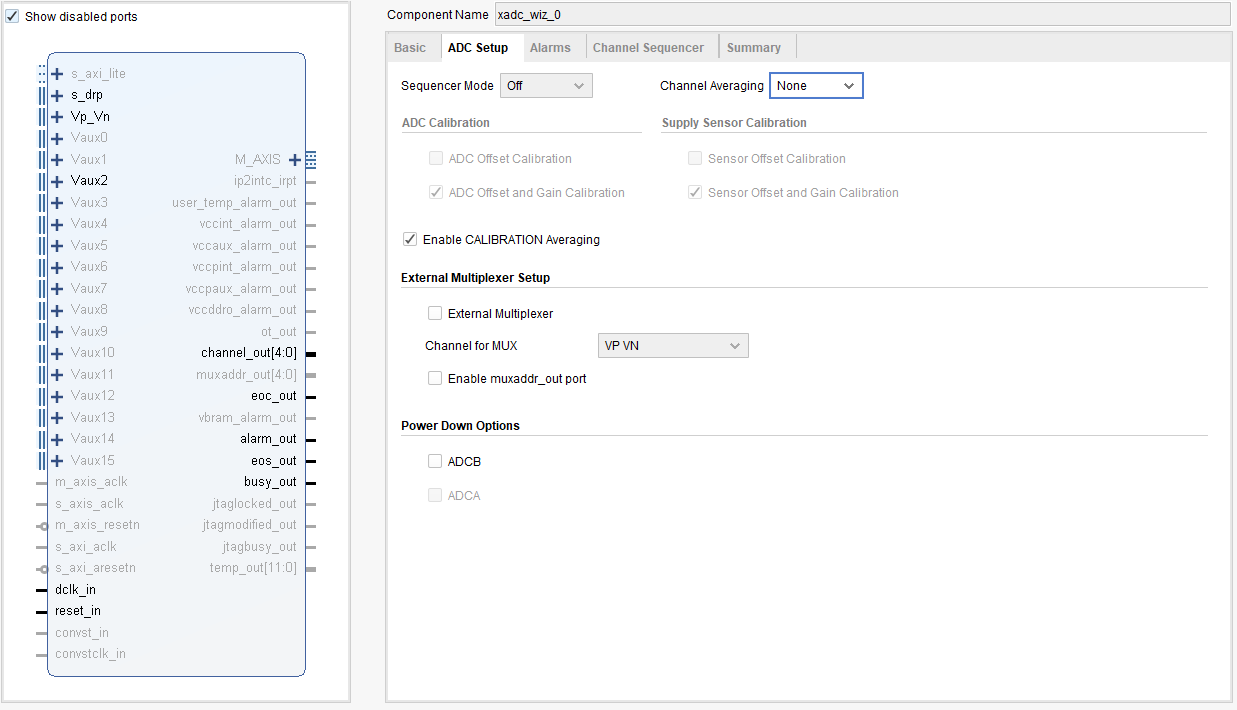


Figure 2b: XADC IP settings 2

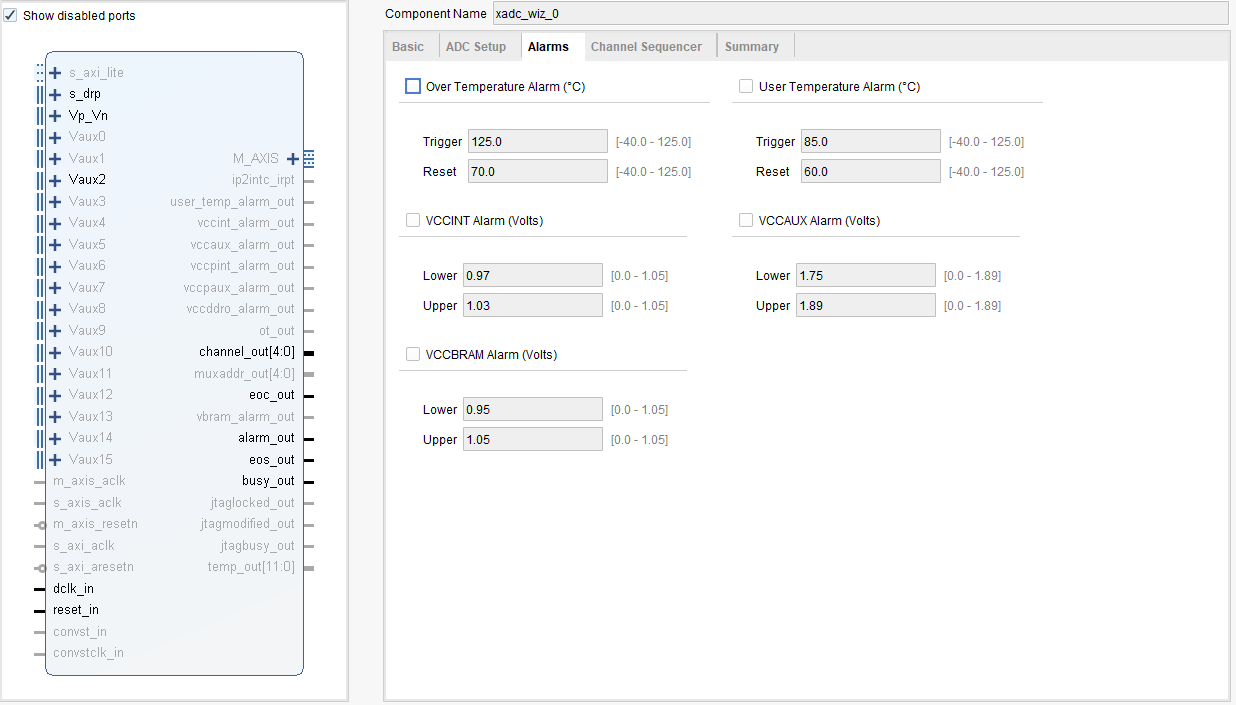


Figure 2c: XADC IP settings 3

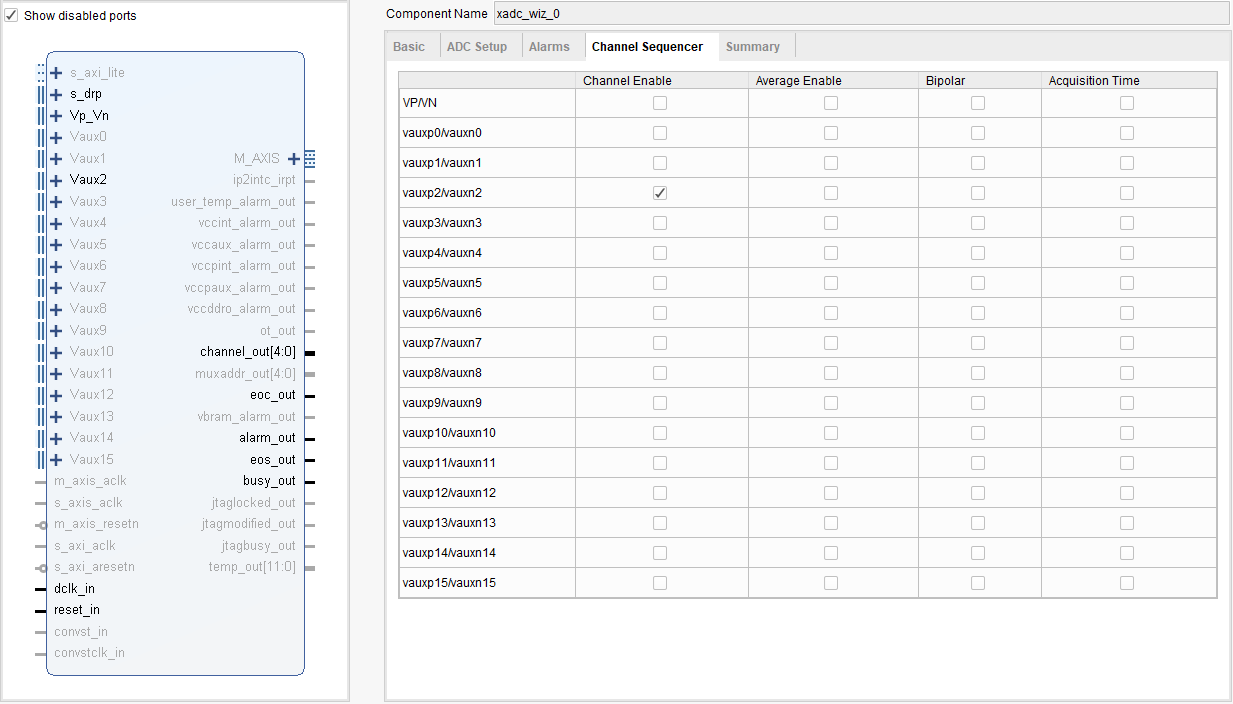


Figure 2d: XADC IP settings 4

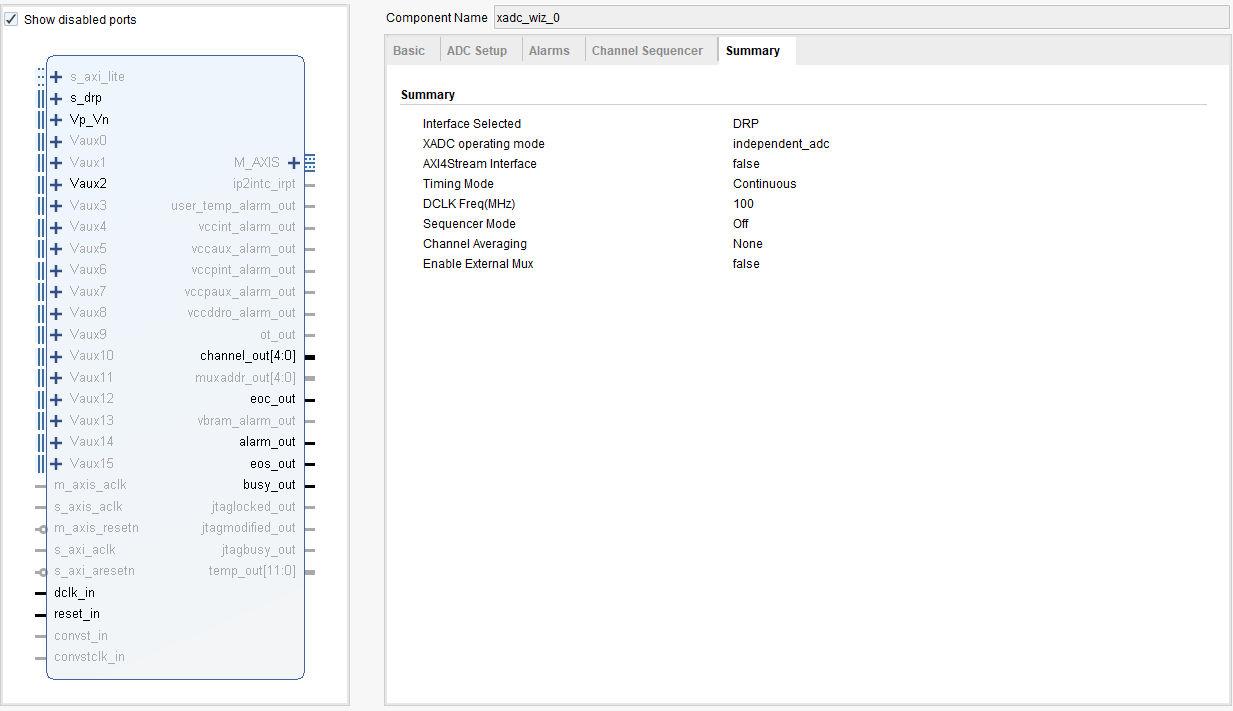


Figure 2e: XADC IP settings 4

1. A Lowpass Filter was designed in Matlab filter designer tool with the following Specifications.

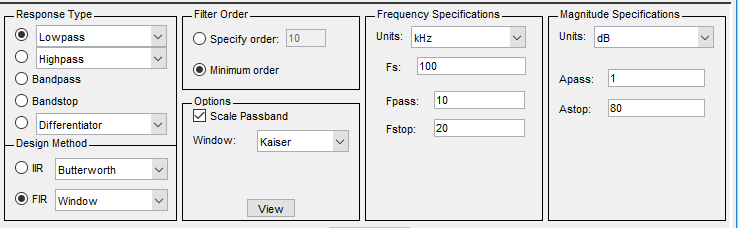


Figure 3a: Filter Specifications 1

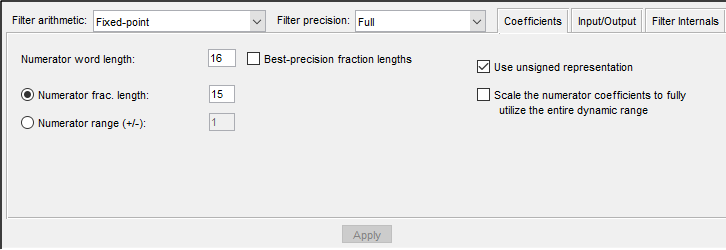


Figure 3b: Filter Specifications 2

1. Once the Filter was designed the coefficients of this filter were exported to Matlab workspace.
2. The coefficients were then wrriten into a csv file using csvwrite(‘filename.csv’,coeffiecientMatrix);
3. The csv file was then opened in matlab as text. Then the coeffiecients were copied from Matlab to be pasted lated on the FIR compiler IP on Vivado.
4. From the IP catalog Fir Compiler IP was instantiated with the following settings.

The copied coefficients from Matlab was pasted into the FIR compiler IP’s coefficient vector.

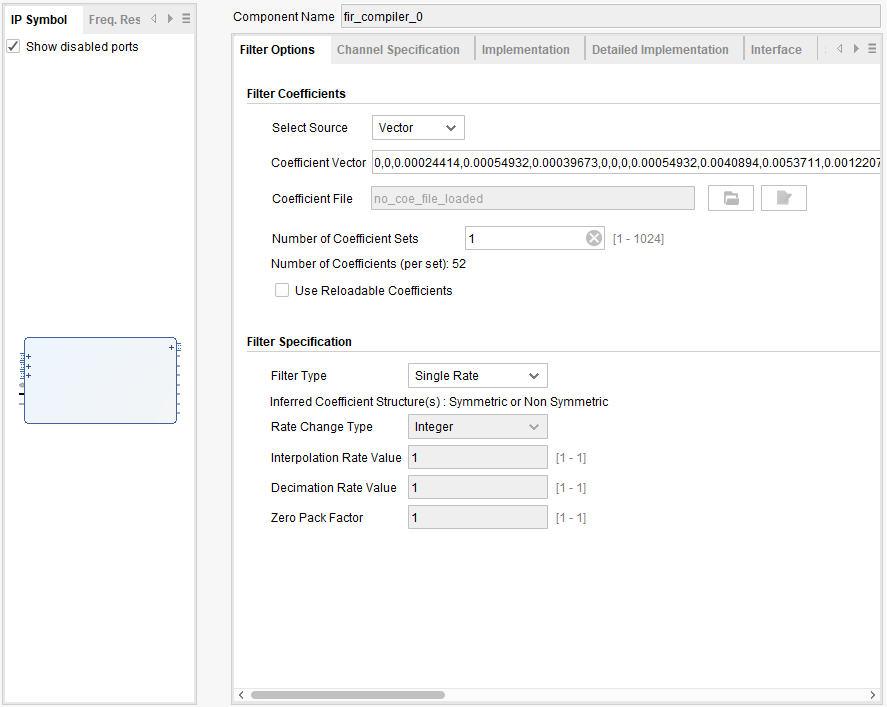


Figure 4a: FIR compiler settings 1

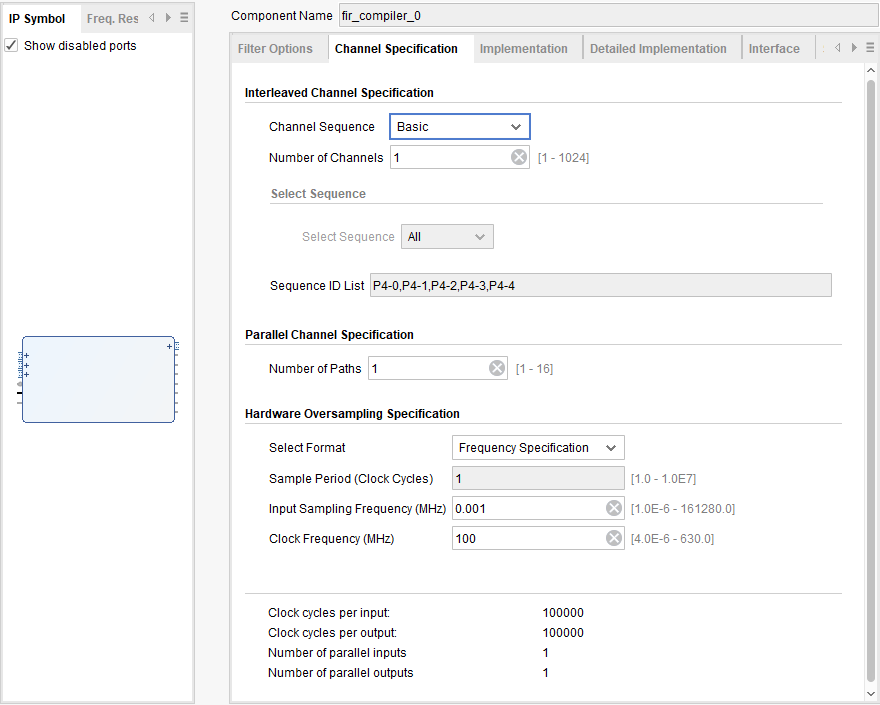


Figure 4b: FIR compiler settings 2

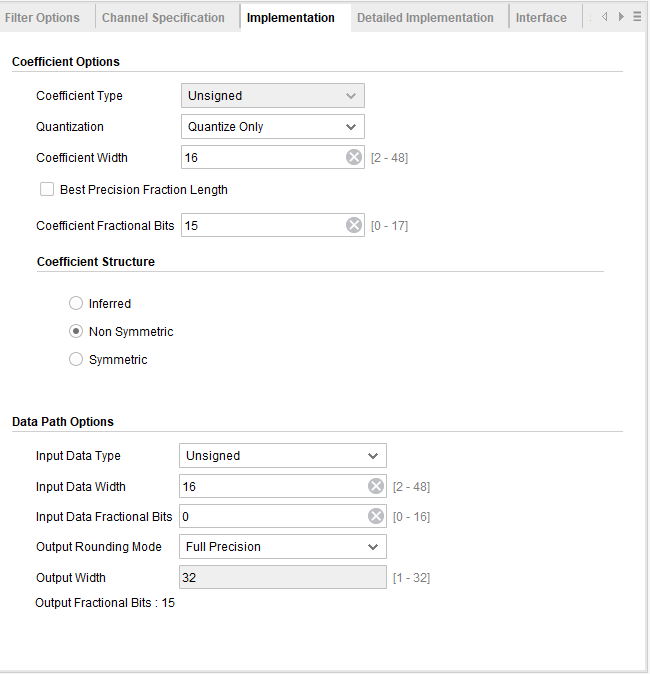


Figure 4c: FIR compiler settings 3

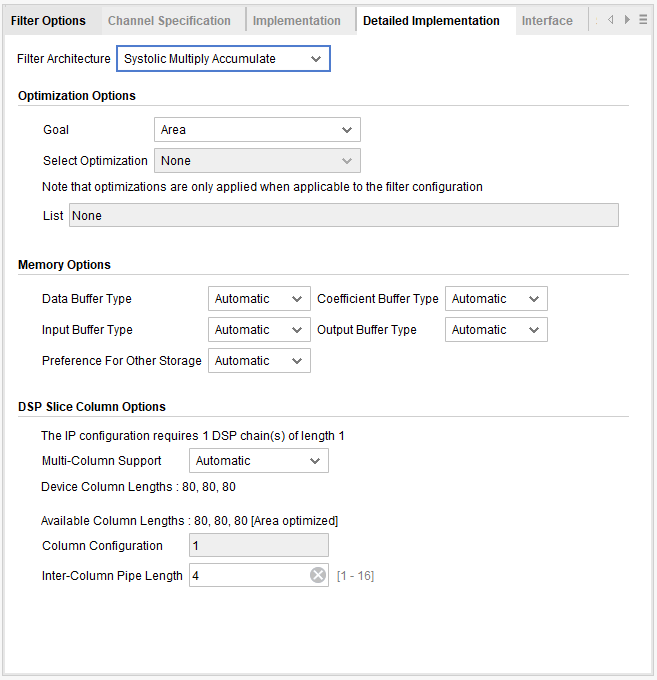


Figure 4d: FIR Compiler Settings 4

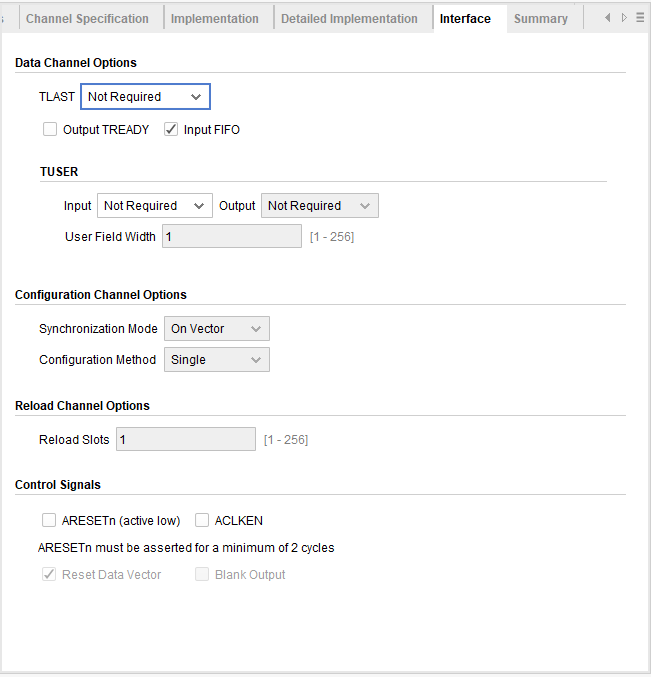


Figure 4e: FIR Compiler Settings 5

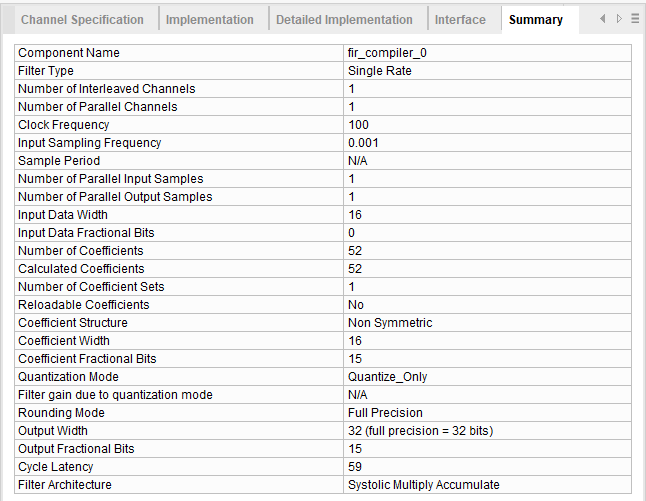
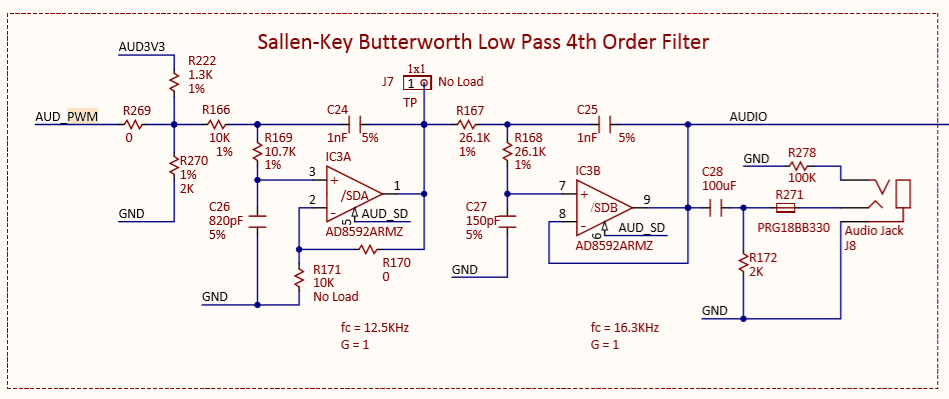


Figure 4f: FIR Compiler Settings 6

1. The XAC was connected to FIR compiler.
2. Then a Pwm code which drives a pin high or low depending on output of the FIR Compiler was written.
3. The PWM signal was then processed using an on board Analog lowpass filter.



[2] Figure 5: NEXYS 4 onboard Sallen-key Butterworth Low-pass 4th order filter.

**Results:**

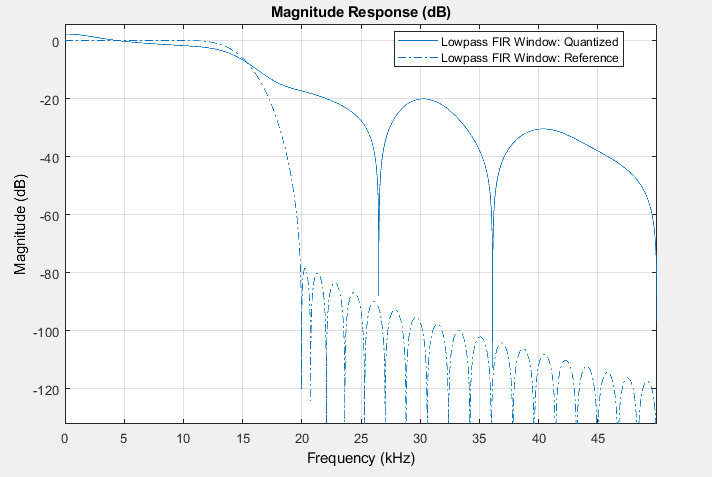


Figure 6: Low-pass Filter Magnitude Response (Matlab)

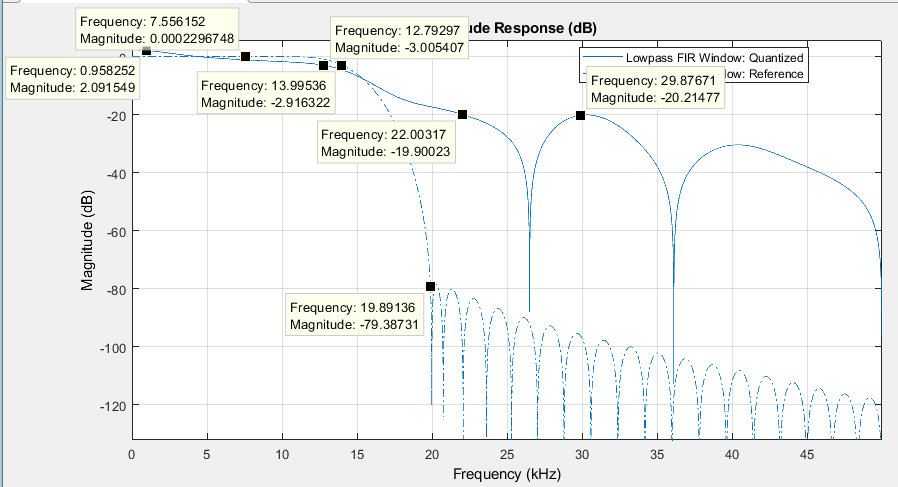


Figure 7: Low-Pass Filter Magnitude Response (Matlab)

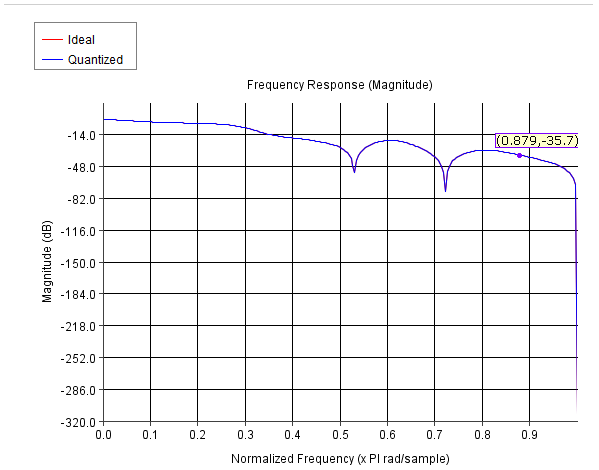
  
Figure 8: Low-Pass Filter Magnitude Response (Vivado FIR Compiler)

Figure 7 shows that the specified Magnitude Response is very different than Quantized Magnitude Response.

Note: “~” denotes about.

Specified

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Fs (kHz) | Fpass (kHz) | Fstop (kHz) | Apass (dB) | Astop (dB) | Phase | Stable |
| 100 | ~12k | ~20k | ~-1 | ~-80 | Linear | Yes |

Quantized

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Fs (kHz) | Fpass (kHz) | Fstop (kHz) | Apass (dB) | Astop (dB) | Phase | Stable |
| 100 | ~13k | ~22k | ~-0 | ~-20 | Linear | Yes |

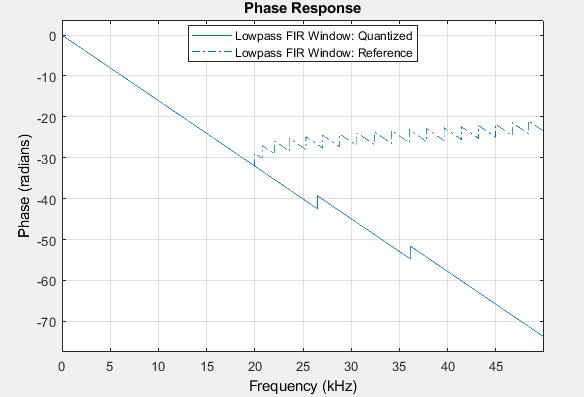


Figure 9: Lowpass Filter Phase Response

Figure 9 shows that the designed filter has Linear Phase

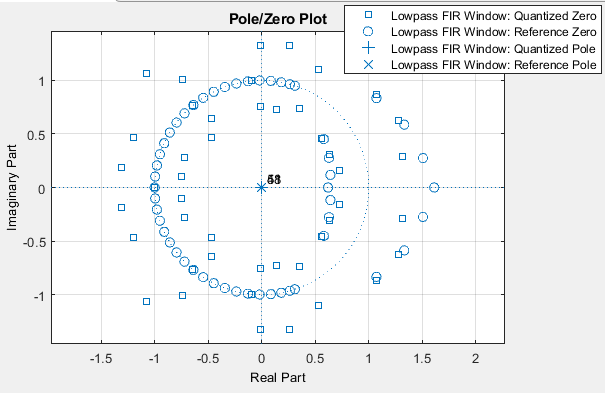


Figure 10: Lowpass Filter Plote Zero Plot

Fireu 10 shows that for both quantized and specified filter are stable as all the poles are inside the unit circle.

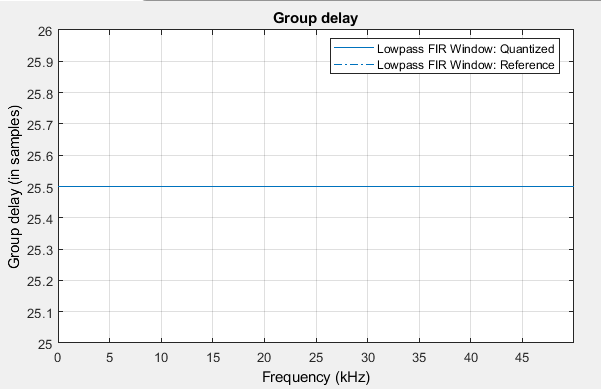


Figure 11: Lowpass Filter Group Delay

Figure 11 shows that the Group delay is constant at 25.5 samples per second. This means that the signal will take some time to propagate through device, which might eventually cause phase shift.

Oscilloscope Traces:

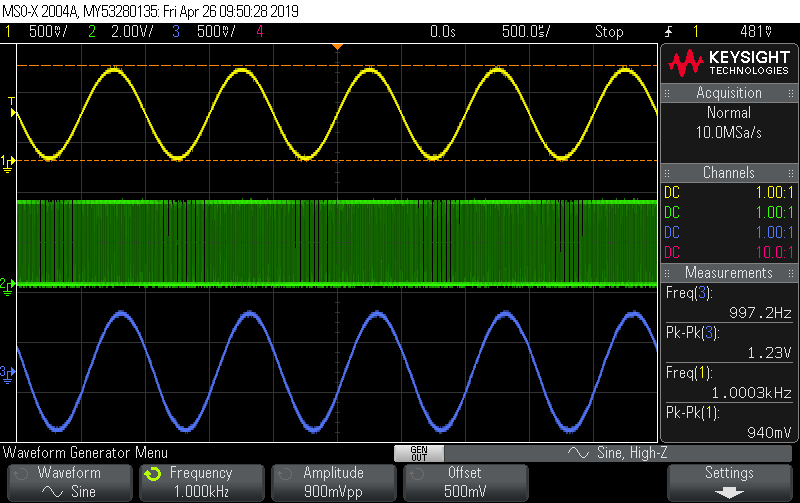


Figure 12: The PWM signal

Figure 12 shows the input signal (Yellow), the pwm signal (Green), the Filtered Signal (Blue). The pwm signal is high everywhere except when input signal is zero.

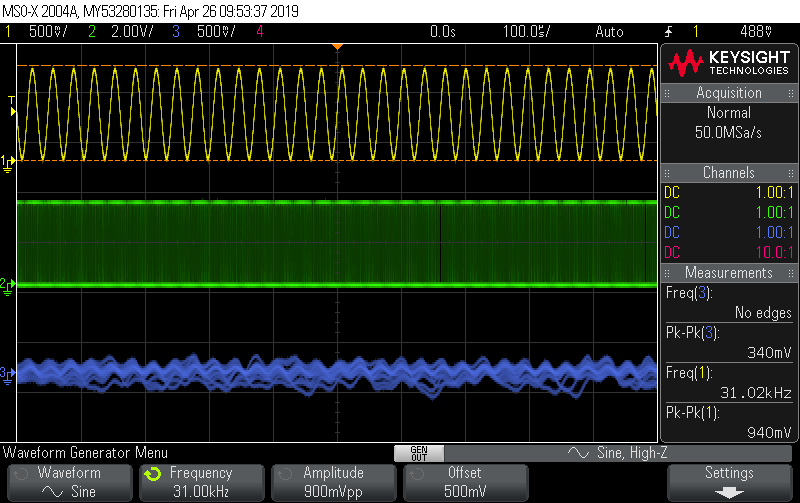


Figure 13

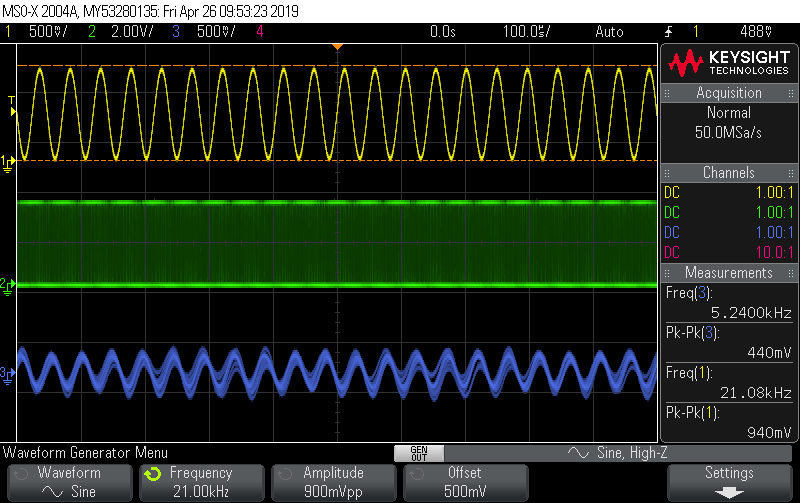


Figure 14

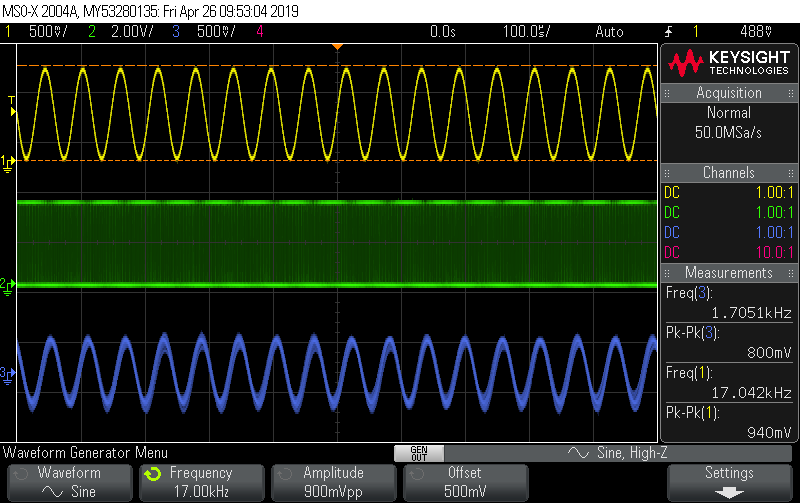


Figure 15

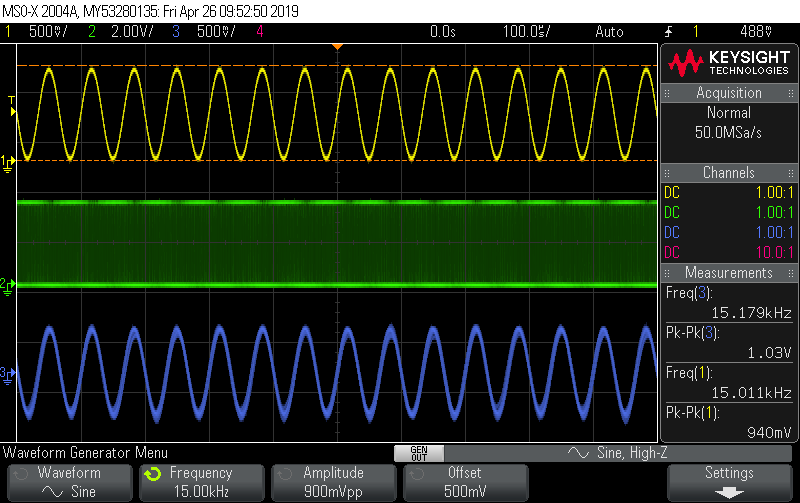


Figure 16

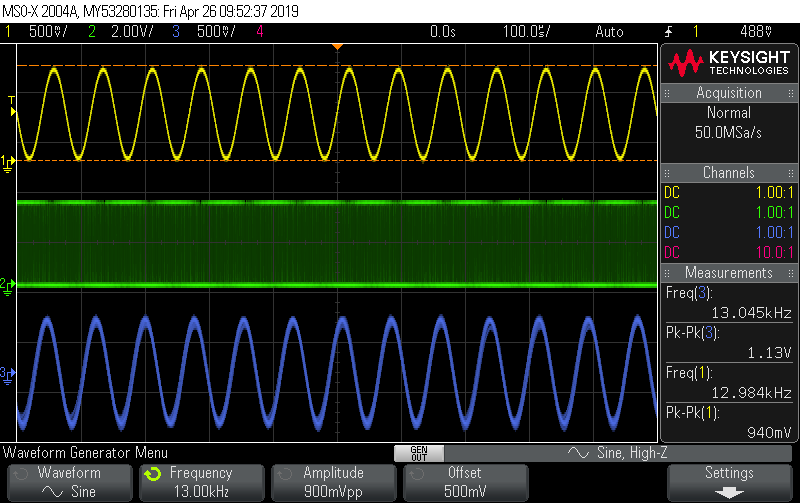


Figure 17

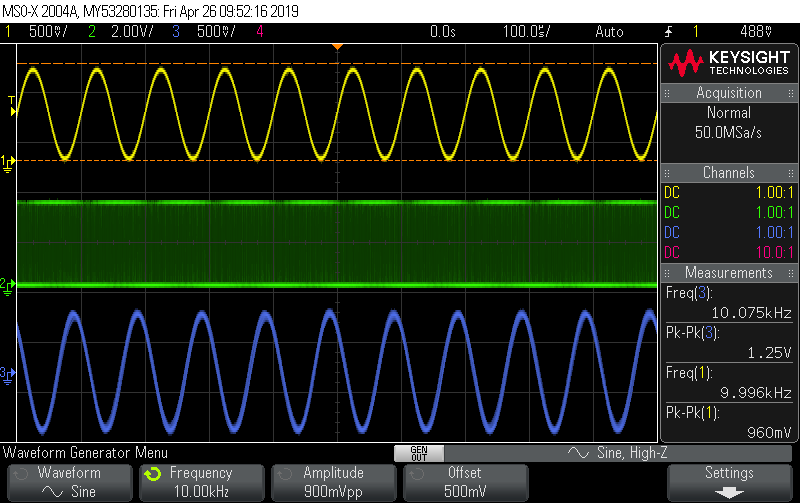


Figure 18

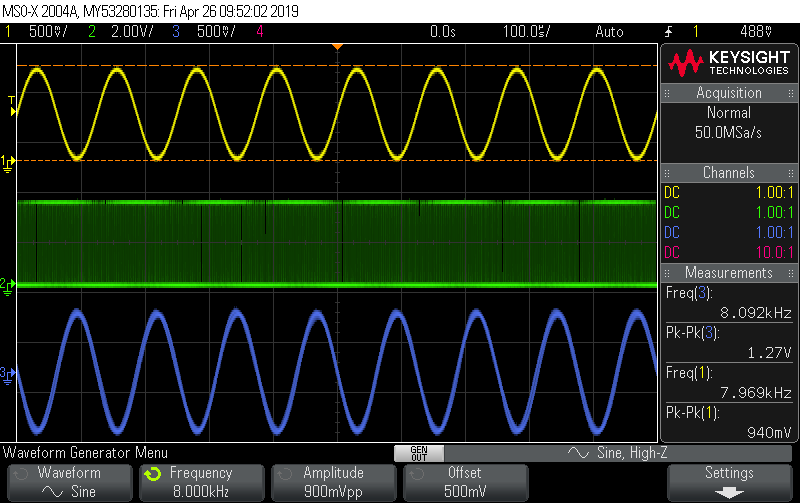


Figure 19

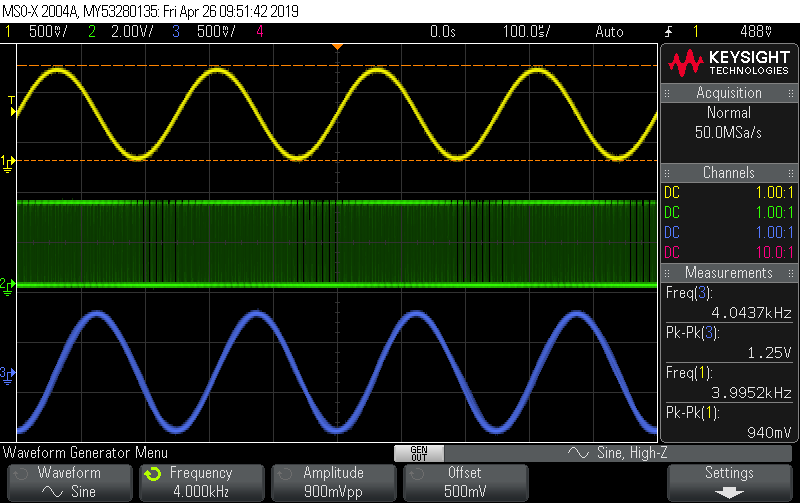


Figure 20

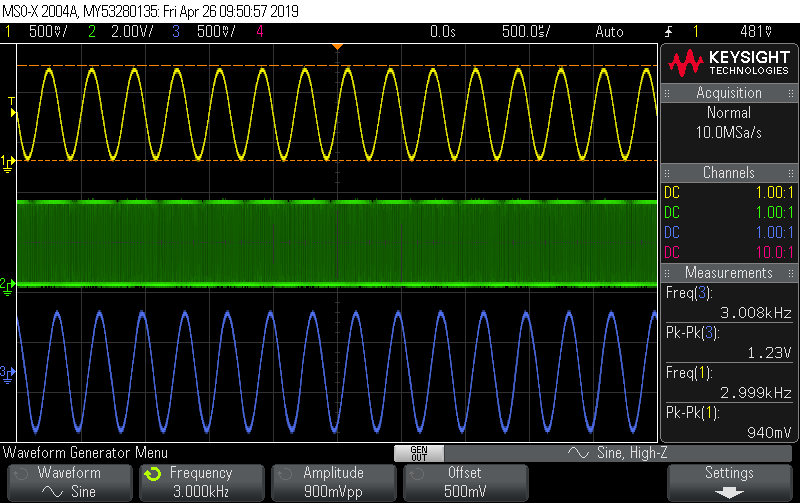


Figure 21

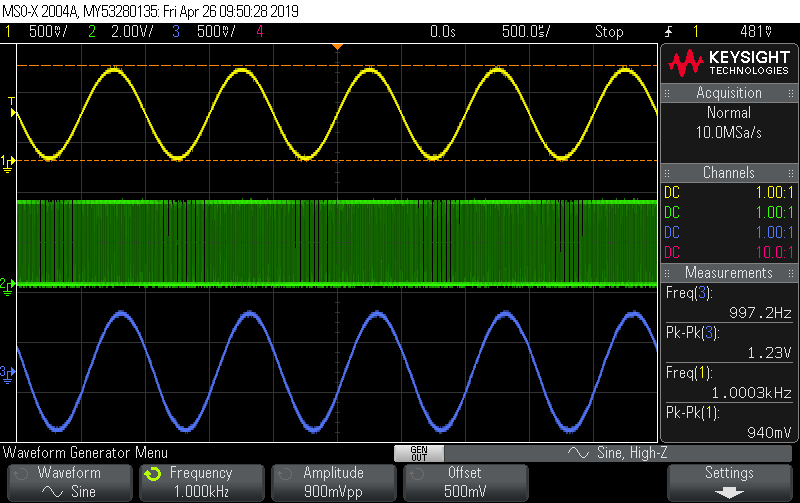


Figure 22

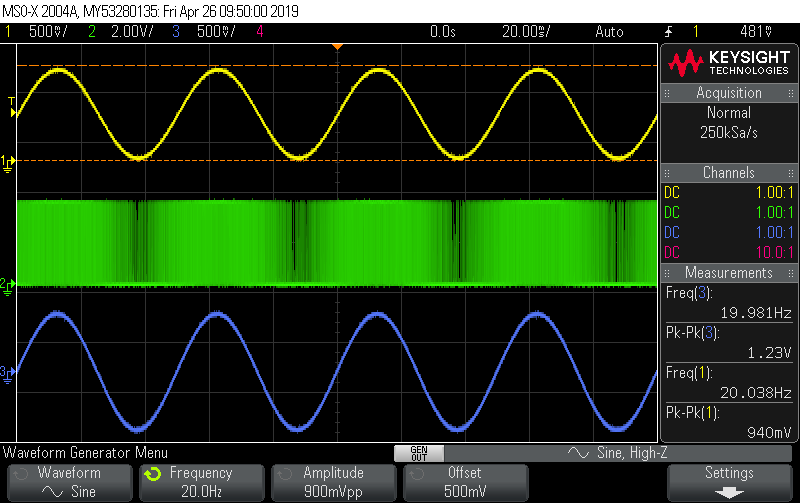


Figure 23

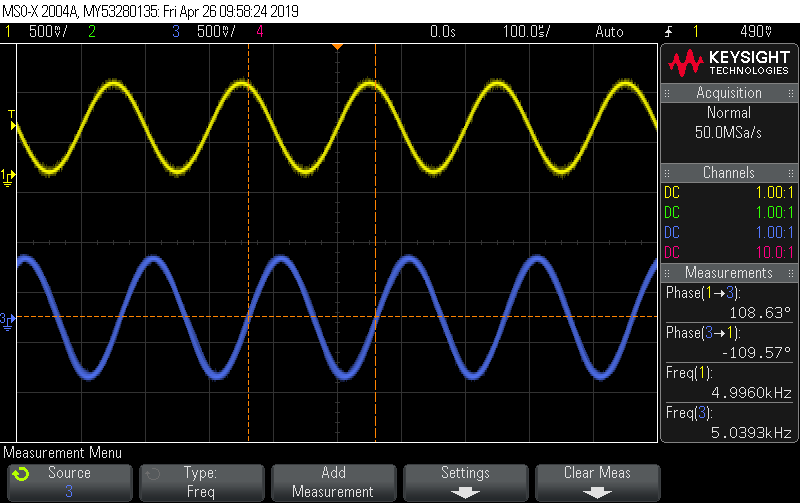
****

Figure 24

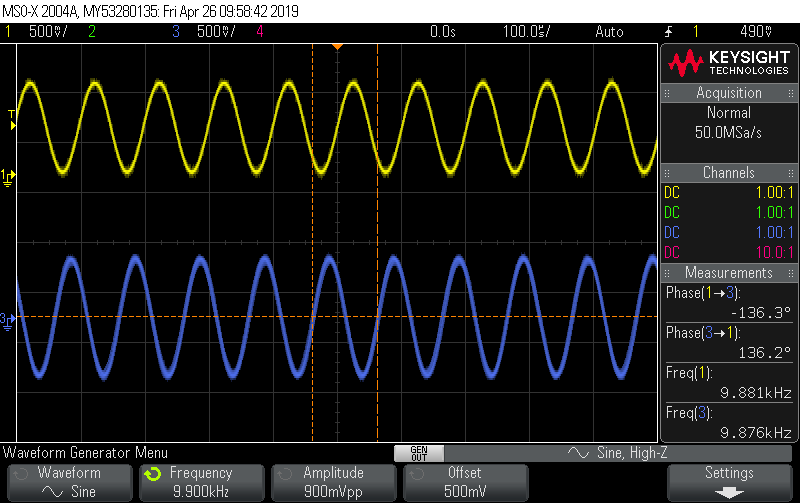
****

Figure 25

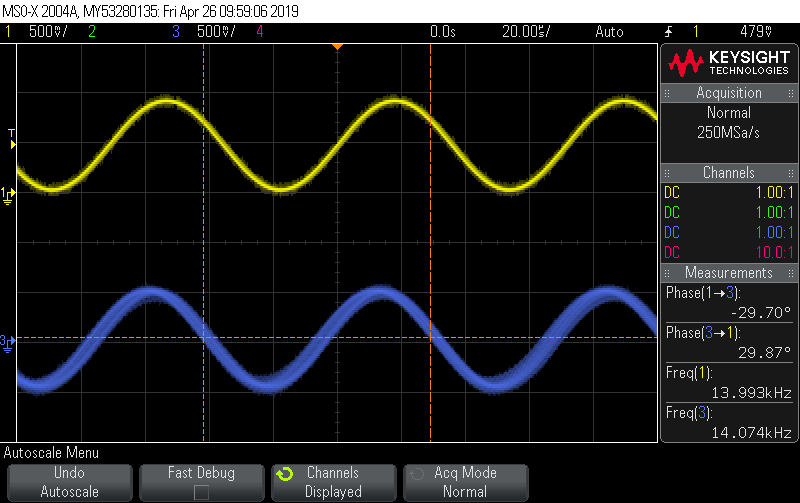
****

Figure 26

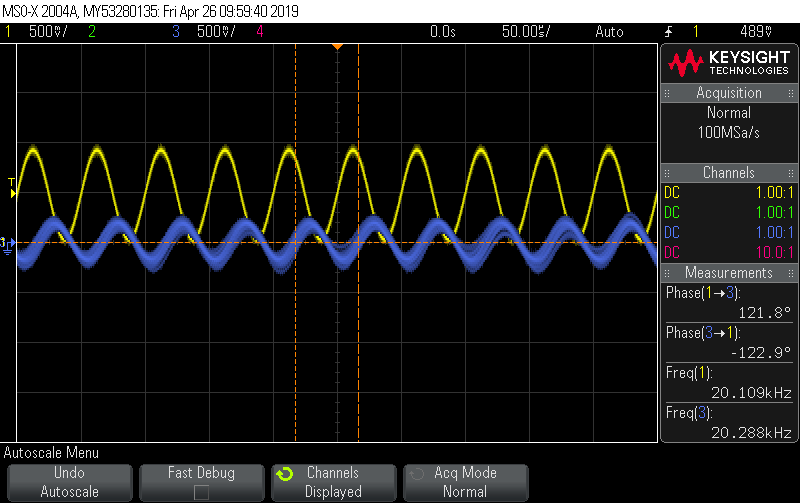
****

Figure 27

In Figure 18-23, the filtered signal(blue) shows that when the input signal(yellow) lies on passband (0-10k) of the implemented low pass filter, the signal passes without any attenuation. The output voltage also seems to be high than input voltage (940 mV).

In Figure 15-17, the filtered signal(blue) shows that when the input signal(yellow) lies on transition band (10K-20k) of the implemented low pass filter, the signal gets more and more attenuated.

In figure 13-14, the filtered signal (blue) shows that when the input signal(yellow) lies on the stopband of the signal is highly attenuated.

Figure 24 – 27 shows that there is some phase delay between input signal (yellow) and the filtered signal (Blue). This delay might have been because of the group delay. The processing device takes some time to process the input signal and output the filtered signal, thus causing delay.

**References:**1) Dr Ling Hou, Class Notes

2) Dr Ling Hou, Project Description

**Verilog Code:**

