ABSTRACT

Digital signal processing finds its application in various modern technologies: audio processing, image processing, space technology, medicine etc. With increasing of computing power increases complexity of task which can be solved. One of the most common instruments of digital signal processing are digital signal processors which can perform different kinds of operations and can be easily programmed.

The purpose of this project is to build audio equalizer on DSP with user interface.

1 TECHNICAL BACKGROUND

1.1 Digital signal processing

Digital signal processing (DSP) is a method of signal processing based on numerical methods using digital computer technology. DSP applications include audio and speech processing, sonar, radar and other sensor array processing, spectral density estimation, statistical signal processing, digital image processing, data compression, video coding, audio coding, image compression, signal processing for telecommunications, control systems, biomedical engineering, and seismology, among others.

Typical DSP scheme is illustrated in Figure 1.1.

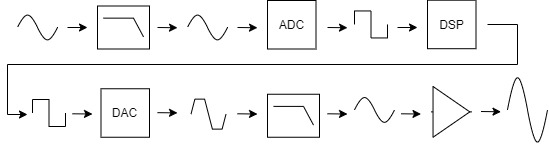


Figure 1.1 – Typical DSP scheme

From the very beginning analog signal should pass through the low-pass filter (LPF) which acts as an anti-aliasing filter. Anti-aliasing filter (AAF) is a filter used before a signal sampler to restrict the bandwidth of a signal to approximately or completely satisfy the Nyquist–Shannon sampling theorem. If not use AAF, aliases of high frequencies (half of Nyquist frequency) will be present in main frequency domain.

Then filtered analog signal goes to analog-to-digital converter (ADC) where it will be discretized (sampled) for further processing in digital processor.

Next stage is a digital signal processor (DSP). It is a specialized microprocessor chip, with its architecture optimized for the operational needs of digital signal processing. It processes data which represents analog signal in such way that is needed for particular project.

Processed signal then goes to digital-to-analog converter in order to become analog again. As it not smooth enough it goes through another low-pass filter which is called reconstruction filter. Reconstruction filter is also needed to prevent imaging – opposite phenomenon to aliasing.

At the last stage signal goes to amplifier in order to be big enough for further processing.

1.2 Digital filters

The relationship between input and output of digital system can be expressed with the following difference equation[1, 60]:

, (1.1)

where y(n) and x(n) – output and input signals respectively;

bi, 0 ≤ i ≤ M and aj, 1 ≤ j ≤ N – coefficients of the system;

n – time index.

Process of computation system output for given input is called digital filtering.

Every system (linear-time invariant) completely characterized by its impulse response[2, 78]. Example of simple digital system is illustrated in Figure 1.2.

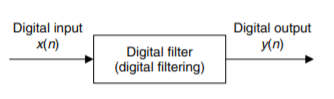
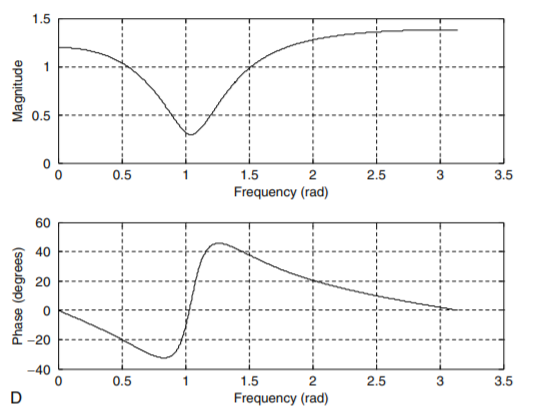


Figure 1.2 – Digital system (Li Tan, Digital signal processing)

One way to describe digital system is its frequency response. Translating impulse response from time domain to frequency domain one can see how system will behave in terms of frequencies and its magnitudes. Example of system impulse response is illustrated in figure 1.3.

Figure 1.3 – Frequency response of digital system (Li Tan, Digital signal processing)

From figure 1.3 it is clearly seen that inputs with frequency 1 rad/sample will become smaller than others.

In other words filter can be explained as system which passes input signals with one particular frequency and stop input signals with other frequencies.

1.3 Types of digital filters

There are two main types of digital filters: finite impulse response (FIR) filters and infinite impulse response (IIR) filters.

1.3.1 FIR filters

FIR filter is a system which can be described by following equation[1, 60]:

, (1.2)

where bi – coefficients of filter;

y(n) and x(n) – inputs and outputs of filter.

It can be seen from equation 1.2 that outputs of FIR depend on present and past inputs.

1.3.2 IIR filters

IIR filter is a system which can be described by equation 1.1. From that equation one can see that outputs of IIR filter depend not only on inputs but also on past outputs. Because of that filter has infinite impulse response.

2 INSTRUMENTS

2.1 Software

2.1.1 MATLAB

MATLAB is a multi-paradigm numerical computing environment and proprietary programming language developed by MathWorks. MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages.

For this project Filter Design & Analysis Tool is used.

2.1.2 Filter Design & Analysis Tool

The Filter Design and Analysis Tool (FDATool) is a powerful user interface for designing and analyzing filters quickly. FDATool enables user to design digital FIR or IIR filters by setting filter specifications, by importing filters from MATLAB workspace, or by adding, moving or deleting poles and zeros. FDATool also provides tools for analyzing filters, such as magnitude and phase response and pole-zero plots.

2.1.3 Proteus

The Proteus Design Suite is a proprietary software tool suite used primarily for electronic design automation. The software is used mainly by electronic design engineers and technicians to create schematics and electronic prints for manufacturing printed circuit boards.

2.2 Hardware

2.2.1 Digital signal processor

For this project PIC microcontroller was chosen as a lot of devices from this series of microcontrollers have built-in DSP core, which makes them suitable for signal processing.

As project is made in simulation some restrictions on the choice of microcontroller are present. The most suitable family of DSP microcontrollers is presented only with few devices.

Selected microcontroller DSPIC33FJ16GP304 has no built-in DAC, so external DAC is used.

2.2.2 Digital to analog converter

For this project MAX5821 was chosen as it can communicate via I2C and has 10 bit resolution.

2.2.3 Other components

Resistors, capacitors, voltage regulators and operational amplifiers are used in this project in order to reach all needed parameters of a circuit.

3 HARDWARE DESIGN

3.1 Primary setups

In order to make microcontroller works properly some primary setups as supplying it with power and etc, should be done.

Considering microcontroller specifications following basic setup circuit was made (fig. 3.1).

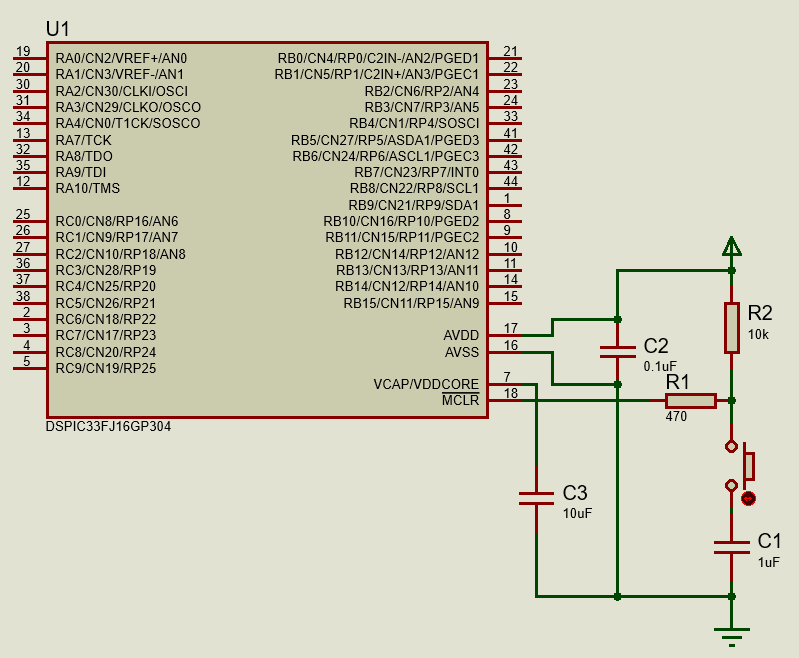


Figure 3.1 – Basic circuit

Maximum voltage of power supply is 3.3V. If circuit will receive power from USB port, voltage regulator is needed as output voltage of USB port is 5V.

For this project LM1117 linear regulator was chosen.

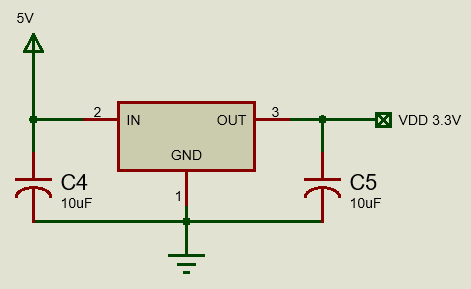


Figure 3.2 – LM1117 scheme

In order to reach maximum MIPS external oscillator is needed. For this purpose 16MHz crystal oscillator was chosen and connected following its datasheet recommendation.

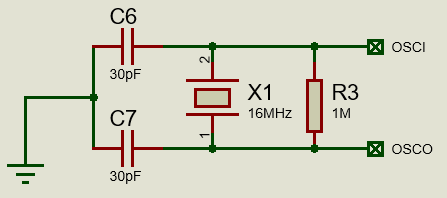


Figure 3.3 – Crystal oscillator connection

3.2 Microcontroller configurations

3.2.1 Oscillator

Microcontroller is configured to operate on 16MHz which produces 40MIPS and clock frequency is 80MHz. Frequency of instructions execution is then 40MHz.

3.2.2 Analog to digital converter

ADC is configured to perform 10-bit resolution and 1.1Msps conversion speed.

As positive reference voltage of ADC is configured to be VDD and negative reference voltage is 0V input signal should have offset. Furthermore, input signal should go through low pass filter to avoid aliasing in further processing. Schematic of signal offset ans filter is illustrated in figure 3.4.

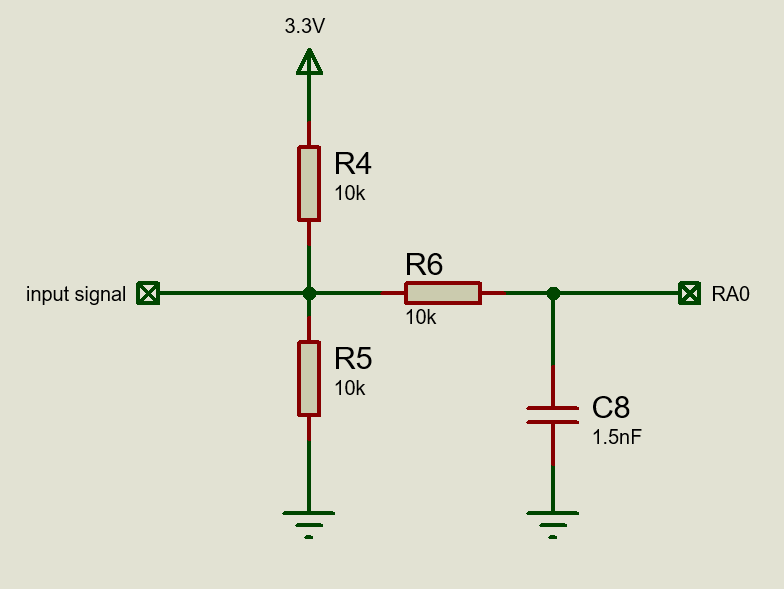


Figure 3.4 – Input signal schematic

Choice of LPF resistor and capacitor will be explained later.

It should be mentioned that input signal will go into circuit from computer’s line-out which has minimal voltage level 300mV and output impedance of 600Ω.

3.2.3 I2C interfaces

In order to communicate with external DAC I2C is needed. As maximum SCL frequency of DAC is 400kHz microcontroller’s I2C was also configured to operate on 400kHz in order to match specifications.

3.2.4 UART interface

In order to convert signals from a RS-232 serial port to signals suitable for use in TTL-compatible digital logic circuits MAX232 is used. Connection schematic is shown in figure 3.5.

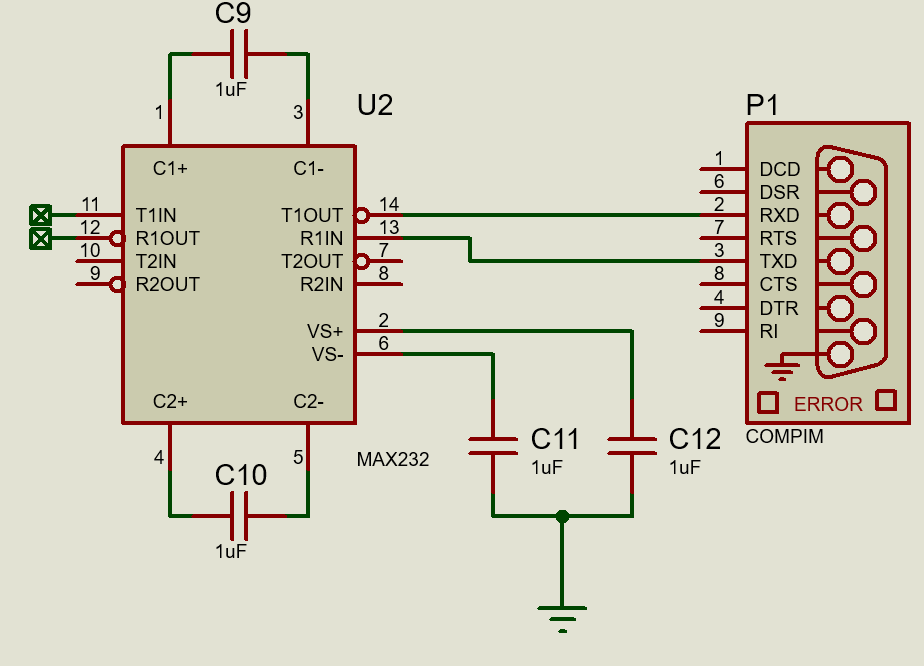


Figure 3.5 – MAX232 connection

3.2.5 External DAC

For this project MAX5821 DAC is used. It has I2C with 400kHz SCL frequency and 10-bit DAC resolution.

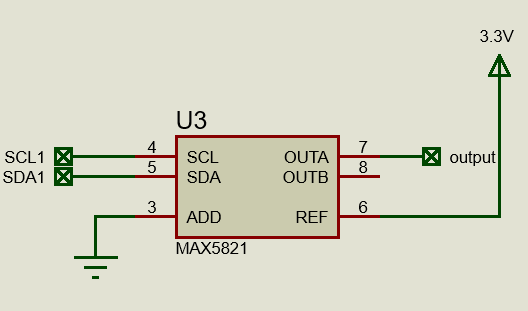


Figure 3.6 – MAX5821 connection

3.2.6 Reconstruction filter

In order to make signal smooth reconstruction filter is needed. Cutoff frequency of such filter would be 9.0kHz as I2C communication produces a huge delay, so in order to be able to filter sound with 20kHz frequency another microcontroller with builtin high speed DAC should be selected. As mentioned previously, this project has such restrictions because of absence of suitable microcontroller in Proteus.

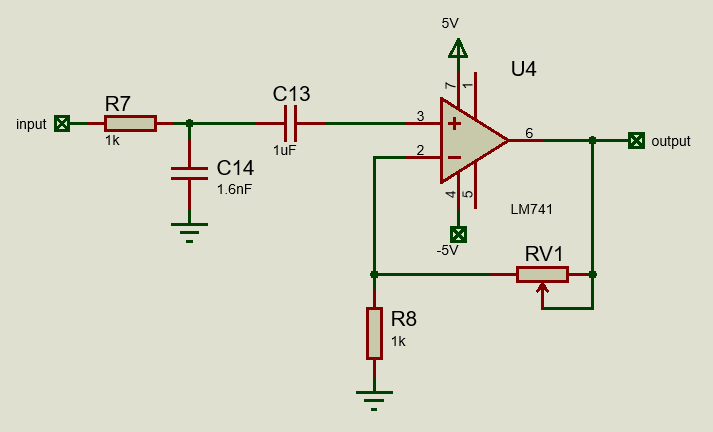


Figure 3.7 – Reconstruction filter with amplifier

Such circuit with potentiometer value 4.7kΩ can amplify voltage up to 4.7 times.

3.2.7 Amplifier negative voltage supply

In order to produce negative voltage supply to operational amplifier voltage regulator P7805 is used.

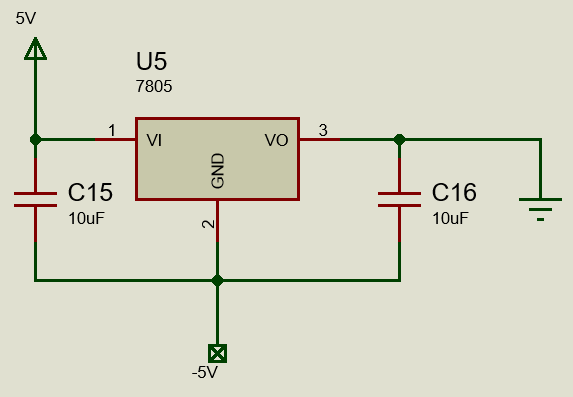


Figure 3.8 – Negative voltage supply

4 SOFTWARE DESIGN

4.1 Audio equalizer

A simplify scheme of audio equalizer is illustrated on figure 4.1.

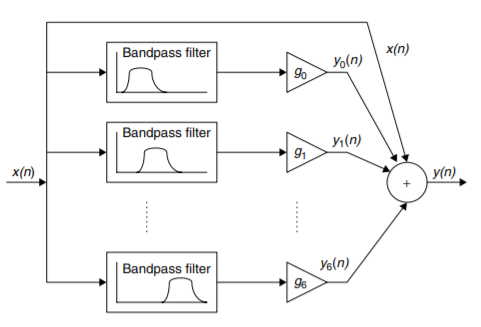


Figure 4.1 – Simplify scheme of audio equalizer

From figure 4.1 one can see, that equalizer consists of several filters, which are combined on one node. Each filter output has its gain, which allows to control frequency response of the equalizer.

Specifications for equalizer:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Center frequency | 500Hz | 1000Hz | 3000Hz | 6000Hz | 9000Hz |
| Bandwidth | 500Hz | 1000Hz | 1500Hz | 2000Hz | 2000Hz |

4.1 FIR filter

For this project FIR filter is used.

FIR filter is calculated with Filter Design and Analysis Tool. Fir filters are designed using window method. For this project Blackman window is used as it has low passband ripple and big stopband attenuation.

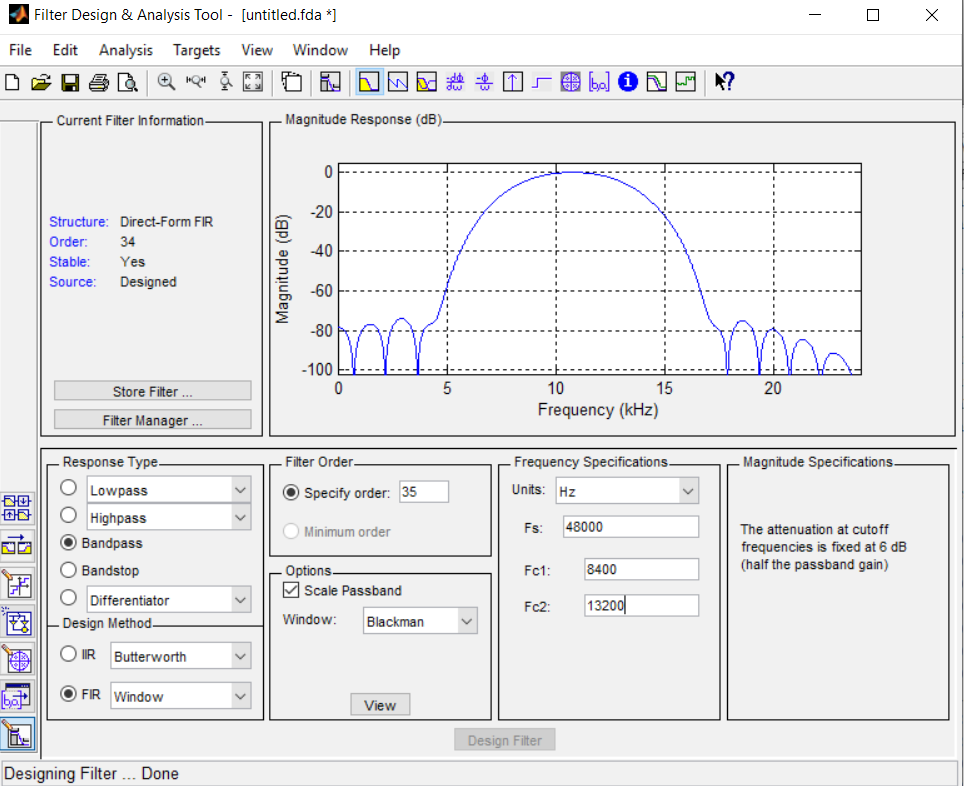


Figure 4.2 – Example of filter design

For this project dsp.h library is used to allow fast computation of filter on embedded DSP core.

5.1 User interface

In figure 4.3 user interface is illustrated.

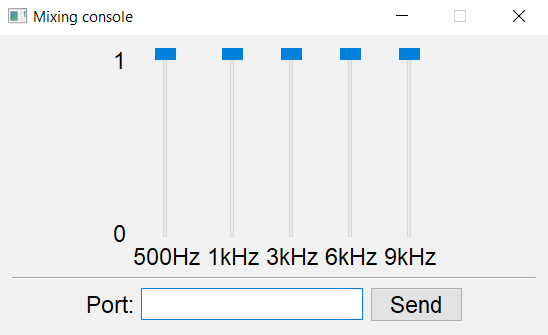


Figure 4.3 – User interface

User interface is written in Python and allows to communicate with microcontroller via UART interface. With this interface user can change magnitude of particular frequency.

5 TESTING

5.1 FIR filters

In this section FIR filters will be tested with builtin Proteus Fourier Analyser.

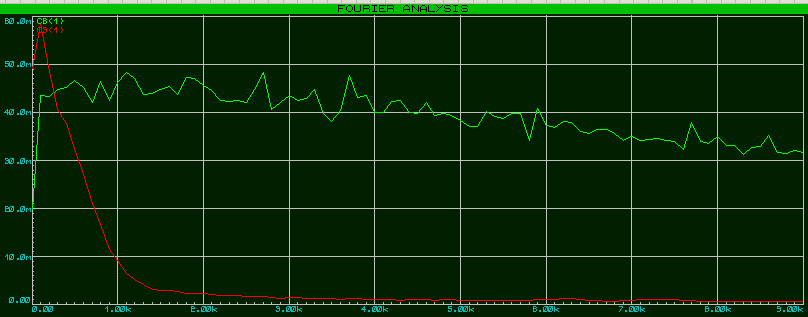


Figure 5.1 – FIR passband filter 500Hz

From figure 5.1 one can clearly see filter in action. Green plot represents white noise and red plot represents filtered signal. Figures 5.2 – 5.6 represent the same.



Figure 5.2 – FIR passband filter 1000Hz



Figure 5.3 – FIR passband filter 3000Hz



Figure 5.4 – FIR passband filter 6000Hz

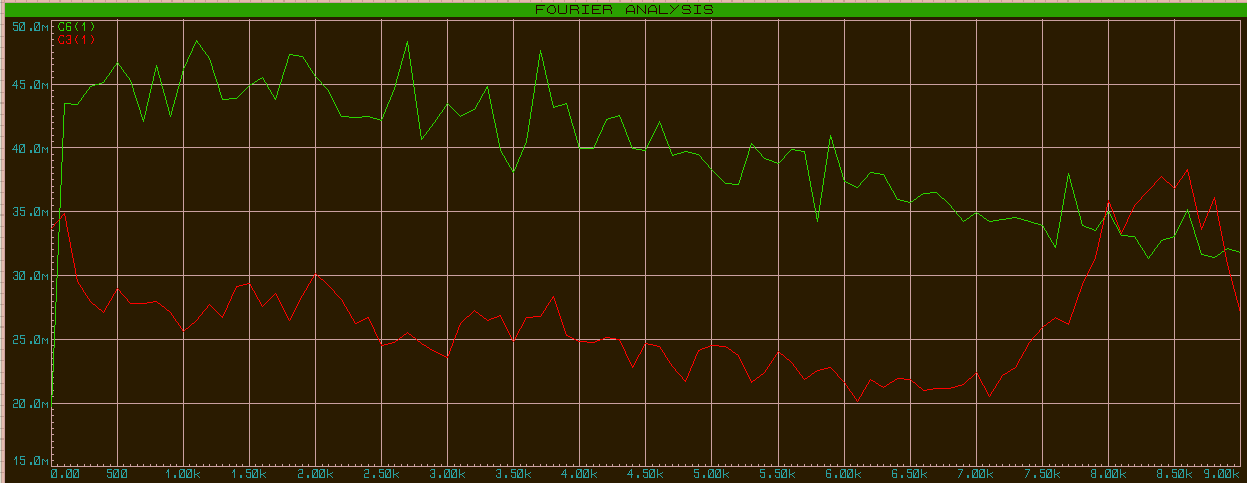


Figure 5.5 – FIR passband filter 9000Hz

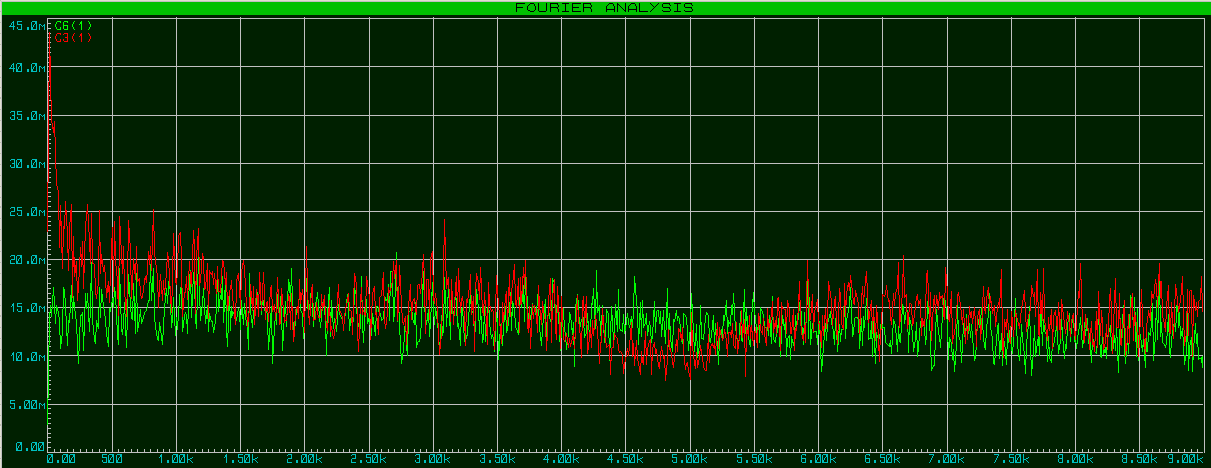


Figure 5.6 – All filters enabled

6 BIBLIOGRAPHY

1. Li Tan, Digital Signal Processing

2. Alan V. Oppenheim, Signals and Systems