

## 1. Brief description of the designed FIR filter and its desired effect.

To Design a filter for the given voice signal, we will firstly analyse where is the noise and where is the target human voice in the frequency domain.

We can find that when the frequency is 93.75, we can obtain the max value in the frequency domain as 8.08.

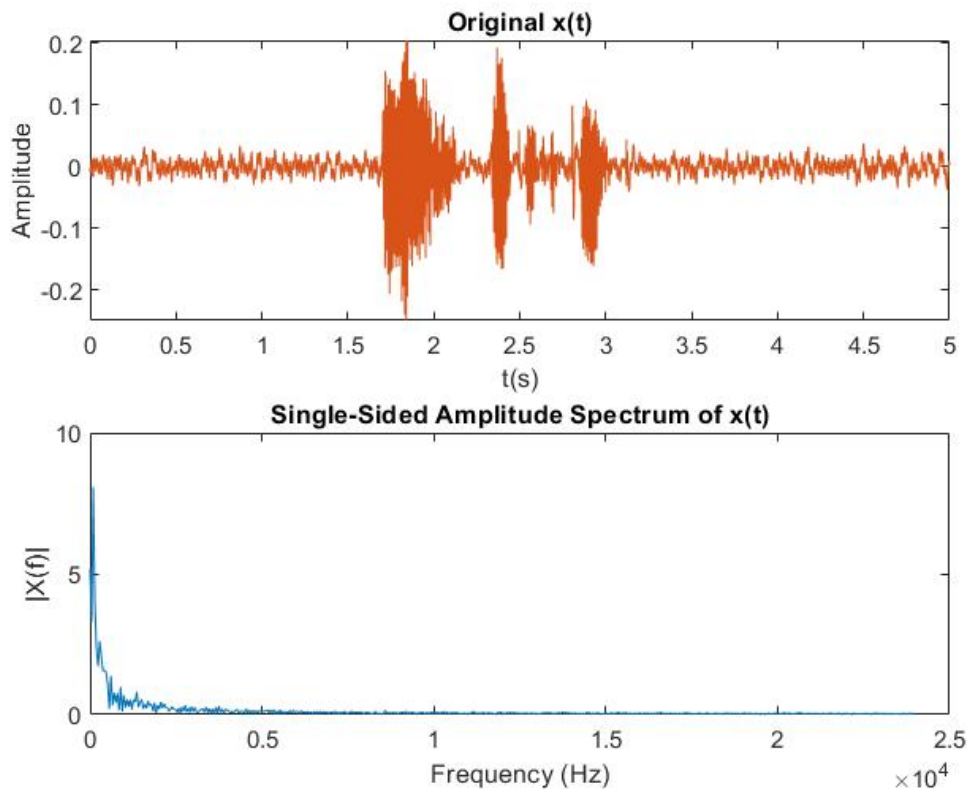


Figure 1. The frequency domain of  $x(t)$

Considering the frequency distribution of human voice ranges from 500 to 1000 and the noise is a low frequency noise. We could filter the original signal by using band-pass filter or a high-pass filter. In order to reduce the consumption of the resource and obtain reasonable performance.

Firstly I have tried to build a high pass filter with a lower order than band pass filter, but it is only performs good in matlab, not so good on the board and I think the SNR of the result could be better if I use band pass filter and the result is stored as “backup\_highpass\_Changhong” ipynb notebook(just a back up to show the difference).

Finally I choose the band pass filter and the parameters are as below.

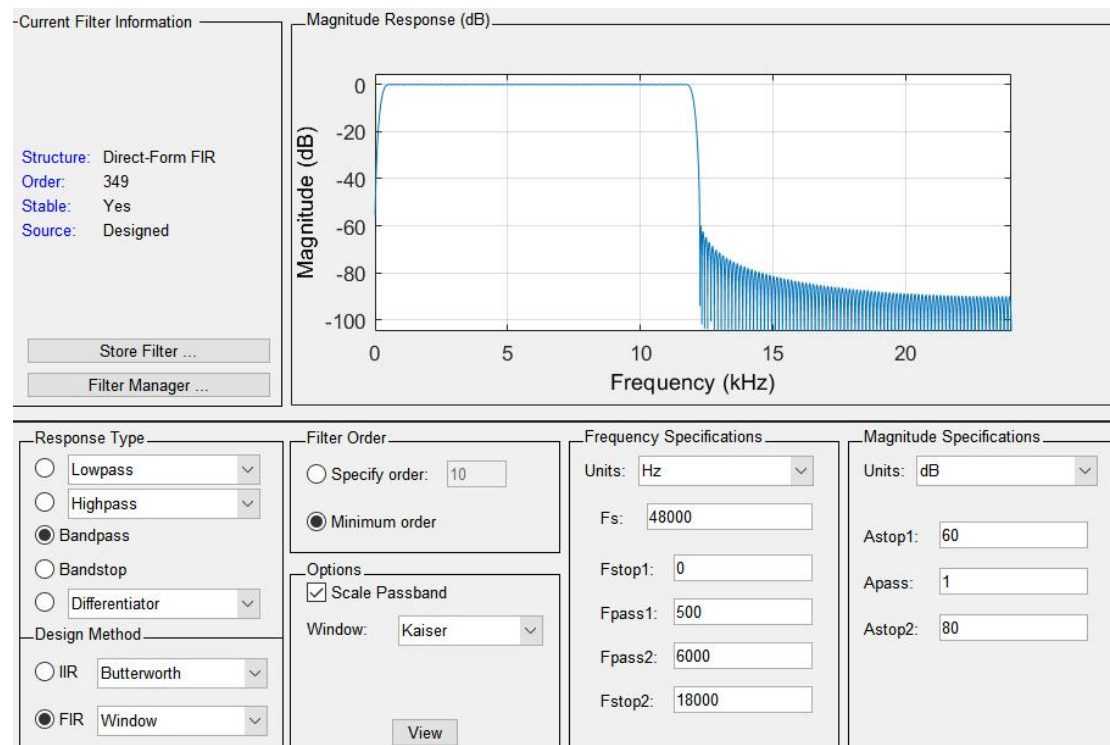


Figure 2. The parameter of the filter

The result waveform is shown as below, we can see that the low frequency noise is removed to a certain degree. Ideally, the filter can remove most of the low frequency noise in this scene, and will not have a large impact on the human voice signal.

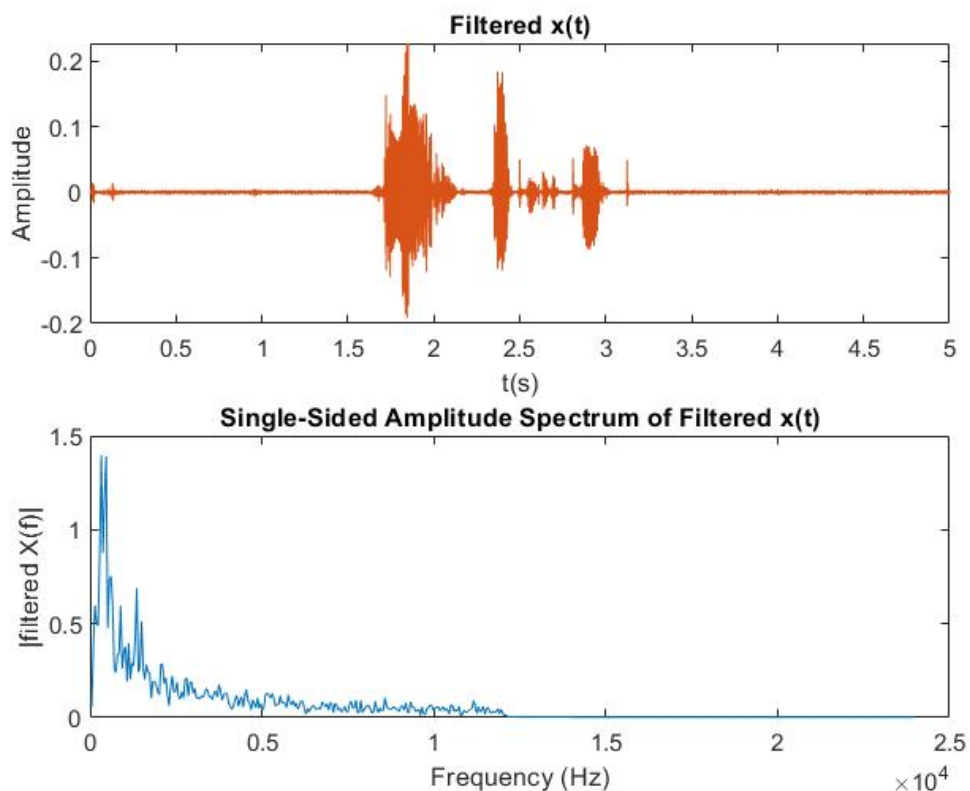


Figure 3. The result of the filter

After we finished the parameter design, we can do the quantization and then transform the filter to the coefficient file which will be used in the vivado filter design.

## **2. Brief description of the role of each IP in the overlay.**

- **audio\_codec\_controller:** Audio CODEC controller using Analog Devices CODEC. CODEC stands for “Coder-Decoder”. It could be used to transform the voice signal for playing in this overlay.
- **AXI GPIO:** Advanced eXtensible Interface General Purpose Input/Output (AXI GPIO) core provides a general purpose input/output interface to the AXI interface. In this overlay, it is used for controlling the GPIO on the board like buttons or LED and so on.
- **Clocking Wizard:** The Clocking Wizard creates an HDL file (Verilog or VHDL) that contains a clocking circuit customized to the user's clocking requirements. Used for generating the audio clock.
- **FIR Compiler:** The Xilinx FIR Compiler LogiCORE is a module for generation of high speed, compact filter implementations that can be configured to implement many different filtering functions. Used for generate FIR filter on the board with the input coefficients.
- **ZYNQ7 Processing System:** Arm dual core SOC with ZYNQ FPGA which carries PS and we can configure the parameters of PS in this part.
- **AXI Direct Memory Access:** The Xilinx AXI DMA LogiCORE provides a flexible interface for transferring packet data between system memory (AXI4) and AXI4-Stream target IP.

- **AXI SmartConnect:** The AXI SmartConnect IP connects one or more AXI memory-mapped master devices to one or more AXI memory-mapped slave devices.
- **Concat:** Concatenates up to 32 ports into a single port
- **AXI-Stream Data FIFO:** The AXI4-Stream Data FIFO IP provides the infrastructure to insert buffering between a AXI4-Stream master and slave.
- **Processor System Reset:** Automatically generated and be used for resetting the PS.
- **AXI Interrupt controller:** The LogiCORE IP AXI Interrupt Controller (AXI INTC) core concentrates multiple interrupt inputs from peripheral devices to a single interrupt output to the system processor.

### **3. Graphs of the audio before and after applying the FIR filter.**

The graphs of the audio which before and after applying the hardware FIR filter are as below.

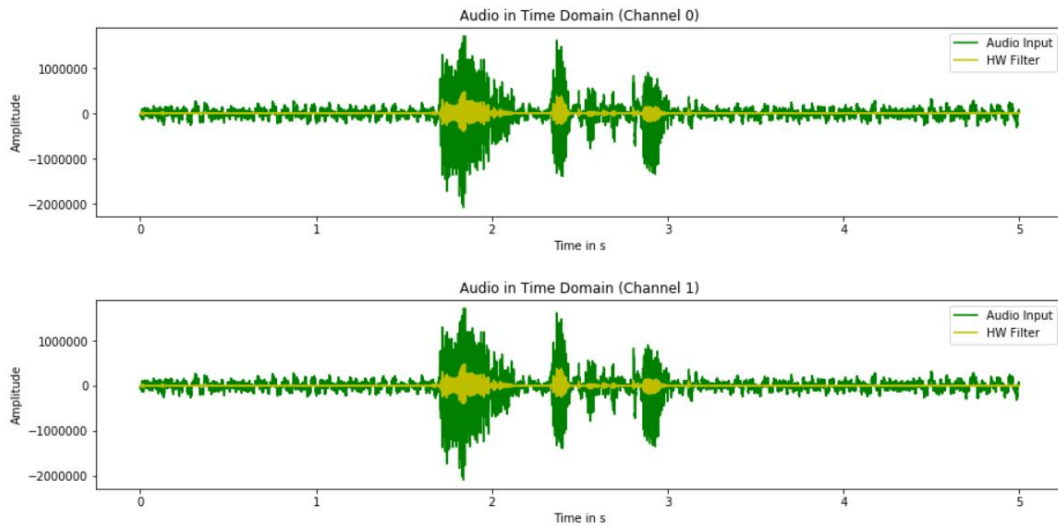


Figure 4. The result of the hardware filter on board

#### 4. Graph comparing the hardware filter output to the software filter output.

The graphs of the audio which before and after applying the software FIR filter are as below. There was no significant difference between the two images.

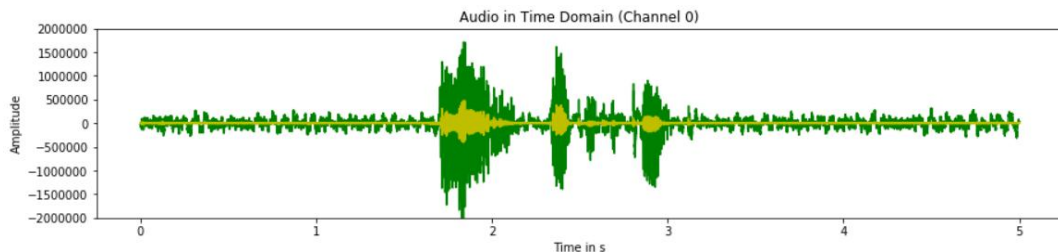


Figure 5. The result of the hardware filter on board

#### 5. Compare the trade-offs between implementing the filter in hardware verses software.

The software and hardware filtering effects of high-pass filter and bandpass filter are as follow table. We can find that the effect of hardware filter and software filter is roughly similar, and the hardware filter

consumes less computing time and resources, but we believe that the software filter will be more flexible. In practical engineering applications, we should make appropriate trade-offs and trade-offs of high-pass filter. In the following table, we can find that the effect of hardware filtering and software filtering is roughly similar, hardware filtering consumes less computing time and resources, but we believe that software filtering will be more flexible, in practical engineering applications, we should be appropriate to make trade-offs.

Table 1. Performance in different cases

	Order	Performance	Hardware time	Software time
BPF	349	Good	0.8408	1.1536
HPF	256	Good but worse SNR on the board than BPF	0.4232	0.9034