FPGA Implementation of Filters

Manas Maurya, Dhruvajit Ghosh

***Abstract*—Digital filters are a very important part of DSP. Filters are used for these two specific purposes – signal separation and signal restoration. In other words, the filter is a device or a process that removes some unwanted components or features from a signal. Our focus is mainly on Finite Impulse Response or FIR filters. Though IIR filters are more efficient, FIR filters are used where precise linear phase is required. Filter implementation on FPGA is preferred in high-performance applications where there is a need to expedite the filtering process. In this paper, we discuss the design and implementation of low-pass, high-pass, and band-pass FIR filters. To obtain the coefficients of the filters, we have used the Windowing technique.**

***Index Terms*— FIR Filter, FPGA, DSP, Windowing Technique**

I. Introduction

Filters are used to suppress some parts of the signal in the frequency domain and give a desired output. This results in the removal of some frequencies or frequency bands. Convolution of the input signal with the digital filter's impulse response gives the output.

An FIR filter is the one whose impulse response is of a finite period, as a result of which it settles to zero in finite time. The impulse response is finite because there is no feedback in the FIR. No feedback guarantees that the impulse response will be finite. The moving average filter where an average of the current input and n-1 past inputs is taken is an example of such a filter.

The output, $y(n)$, of an FIR system which has been given the input, $x(n)$, is represented by following mathematical the equation.

$$y(n) = \sum_{i=0}^{N} b_i \cdot x(n-i)$$

where b_i is the value of the impulse response at the i^{th} instant for $0 \leq i \leq N$ of an N^{th} order FIR filter. If the filter is a direct form FIR filter, then b_i is also a coefficient of the filter. This equation can be realized as a tap delay structure, as shown below.

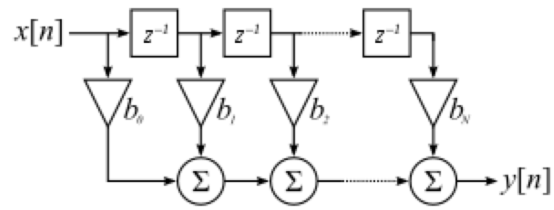

Figure 1. Tap delay structure for FIR

FIR filters can further be divided with respect to what part of the frequency spectrum they allow to pass through them. Our main focus is on three of those types, namely, the low-pass, the high-pass, and the band-pass.

A **low-pass filter** is a filter that passes signal components which lie below a certain frequency, known as the cutoff frequency and attenuates components with frequencies higher than the cutoff frequency. The ideal frequency response of such a filter is as shown below and is followed by the practical frequency response of the same.

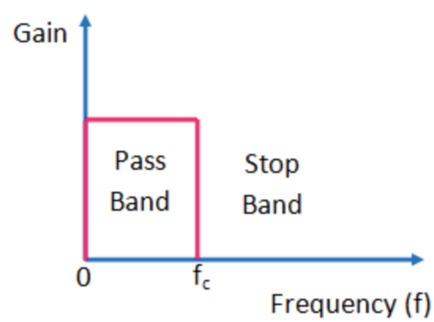

Figure 2. Ideal frequency response of a low pass filter

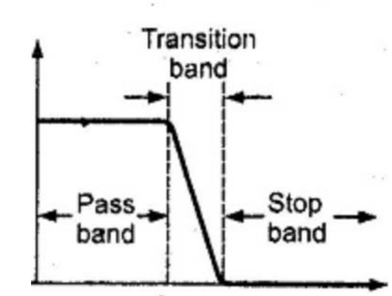

Figure 3. Practical Frequency response of a low pass filter

A **high pass filter** is a filter that passes components of the signals which lie beyond the cut off frequency and attenuates the components which lie below the cut off frequency. The ideal frequency response of the high pass filter is shown below, followed by the practical frequency response of the same.

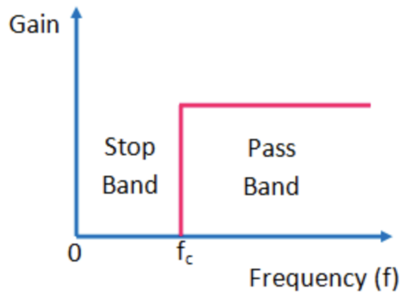

Figure 4. Ideal frequency response of the high pass filter

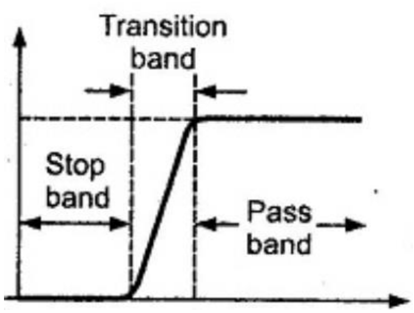

Figure 5. Practical frequency response of the high pass filter

A **band pass filter** is a filter that passes the signal components which lie in a certain range of frequencies defined by two cut off frequencies and attenuates the components lying outside this range. The ideal frequency response of such a filter is shown below, followed by a practical response of the same.

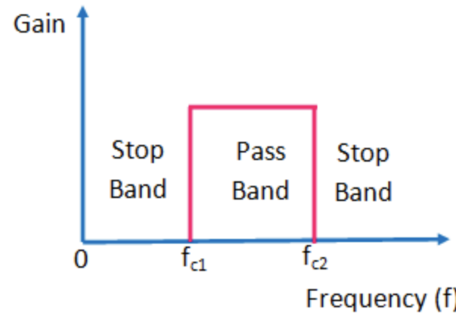

Figure 6. Ideal frequency response of a band pass filter

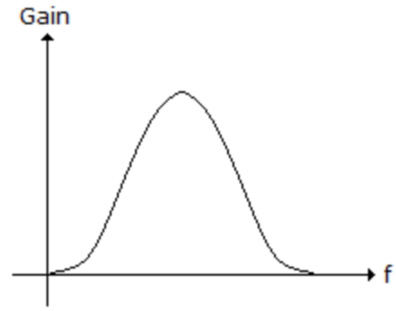

Figure 7. Practical frequency response of a band pass filter

II. PROBLEMS/EXAMPLES CONSIDERED

A. *Low pass filter Example*

Moving average(MA) filter is the simplest representation of a low pass filter, and hence that is the filter that we have considered for implementation. The general equation of a moving average filter is as shown below.

$$y(n) = \frac{1}{N}\sum_{k=0}^{N-1} x_{n-i}$$

where $y(n)$ represents the output and $x(n)$ represents the input.

B. *High pass filter Problem*

In order to implement a high pass filter and a band pass filter, we have taken arbitrary design problems.

Design problem considered for the high pass filter lists the following specifications.

Passband edge frequency: - 498 Hz
Stopband edge frequency: - 200 Hz
Attenuation in stop band: -greater than 40dB
Sampling frequency: - 1200 Hz
Passband ripple: - 0.2dB

C. *Band Pass Filter Problem*

Design problem considered for the band pass filter lists the following specifications.

Passband width: - 300 Hz

TABLE 1. WINDOWING FUNCTIONS

Type of Window	Equation of the window $w[n]$	Peak Side – Lobe Amplitude	Approximate width of Main lobe	Peak Approximate Error, $20\log_{10}\delta$ (dB)	Transition Width of equivalent Kaiser window
Bartlett	$\begin{cases} \frac{2n}{M}, & 0 \le n \le M/2 \\ 2-\frac{2n}{M}, & M/2 < n \le M \\ 0, & otherwise \end{cases}$	-25	$8\pi/M$	-25	$1.81\pi/M$
Hanning	$\begin{cases} .5-.5\cos\left(\frac{2\pi n}{M}\right), & 0 \le n \le M \\ 0, & otherwise \end{cases}$	-31	$8\pi/M$	-44	$2.37\pi/M$
Hamming	$\begin{cases} .54-.46\cos\left(\frac{2\pi n}{M}\right), & 0 \le n \le M \\ 0, & otherwise \end{cases}$	-41	$8\pi/M$	-53	$6.97\pi/M$
Blackmann	$\begin{cases} .42-.5\cos\left(\frac{2\pi n}{M}\right) + .08\cos\left(\frac{4\pi n}{M}\right), & 0 \le n \le M \\ 0, & otherwise \end{cases}$	-57	$12\pi/M$	-74	$9.19\pi/M$

Center frequency: - 500 Hz
Width of the Transition band- 333 Hz
Attenuation in stop band: - greater than 50dB
Passband ripple expected to be less than 0.1dB
Sampling frequency: - 2000 Hz

III. WINDOWING TECHNIQUES

The windowing method is the easiest procedure which we can use to design FIR filters. The method generally begins with an ideal desired frequency response that can be represented as

$$H_d(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h_d[n]e^{-j\omega n}$$

where $h_d[n]$ is the corresponding impulse response sequence, which can be expressed in terms of $H_d(e^{j\omega})$ as

$$h_d[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d\,(e^{j\omega})\,e^{j\omega n}d\omega$$

These systems have impulse responses that are noncausal and infinitely long. So, in order to obtain a causal and a finite response, we truncate this response using a finite duration function called the window function. The use of the simplest window function, the rectangular function, is as shown below by the following equations.

$$h[n] = h_d[n]\,w[n]$$

$$w[n] = \begin{cases} 1, & 0 \le n \le M \\ 0, & otherwise \end{cases}$$

DTFT of h[n] is given by

$$H(e^{j\omega}) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(e^{j\theta})W(e^{j(\omega-\theta)})d\theta$$

where $W(e^{j\omega})$ is the DTFT of the window function.

The rectangular window is not used frequently due to the problem caused by the increase in the side lobe amplitudes (in the magnitude plot of the Fourier transform $H(e^{j\omega})$) with the increase in the order of the filter. The side lobe energies for the rectangular window are very high, even at lower orders. This is due to the sharp cut off in $W(e^{j\omega})$. To resolve this issue, several other windows were introduced, which would taper to zero smoothly. The specifications of such windows with their names have been tabulated above.

IV. CALCULATIONS

Considering the example of the **low pass filter** first, we have the equation that was mentioned earlier: -

For the sake of simplicity, we have decided to use the tap delay structure with only 8 taps. Hence, we will need to calculate 8 coefficients (or 8 b_is).

n	0	1	2	3	4	5	6	7	8
h[n]	0	.0013	.0013	-231/50000	-29/1000	-.071	-.124	-.167	.815

Table 1. h[n] values for High pass filter

n	9	10	11	12	13	14	15	16
h[n]	-.167	-.124	-.071	-29/1000	-231/50000	.0013	.0013	0

$$y(n) = \frac{1}{N}\sum_{k=0}^{N-1} x_{n-i}$$

Since we already have a relation between the input and the output, we can easily deduce the coefficients which need to be multiplied. In this case, we have all the 8 values of the impulse response (or coefficients) to be equal to $\frac{1}{8}$.

Next, considering the problem taken up for the **high pass filter.**

Correlating the specifications given for the design and the data mentioned in Table 1, we conclude that using the Hann window would prove to be the most effective.

We have adjusted the passband and stopband edge frequencies such that we get the length of the window, M, as 16 (or 17 coefficients). An ideal high pass filter with cutoff frequency ω_c has the following impulse response: -

$$h_d[n] = sinc(n) - \frac{\omega_c}{\pi} sinc(\frac{\omega_c n}{\pi})$$

and using the relation $h[n] = h_d[n]\, w[n]$, we have,

$$h[n] = (.5 - .5\cos\left(\frac{2\pi n}{16}\right))(sinc(n) - .581 sinc(.581n)$$

$$0 \le n \le 16$$

similarly, when considering for **band pass filter**, we get the following: -

$$h_d[n] = \frac{\omega_{c,u}}{\pi} sinc(\frac{\omega_{c,u} n}{\pi}) - \frac{\omega_{c,l}}{\pi} sinc(\frac{\omega_{c,l} n}{\pi})$$

$$h[n] = (.54 - .46\cos\left(\frac{2\pi n}{24}\right))(.816 sinc(.816n) - .183 n sinc(.183n))$$

$$0 \le n \le 24$$

Here the number of taps is equal to 25.

V. Use of Verilog HDL

For writing the Verilog code, we have considered the input and the output to be 16 bits each. To make the code simple, we have considered a step input.

A basic Verilog code can be written for any FIR filter while considering the tap delay structure. The main processes involved include multiplication of the coefficients of the impulse response with the tapped signals, the delaying of the input signal by a single delay element before each multiplication except for the very first tap and summing of all the terms obtained after the multiplication.

The delay element can easily be implemented by using a d flip flop. On the other hand, the implementation of the multiplier is not that easy and involves a few approximations.

All the coefficients that have been obtained manually are fractional. We divide both the numerator and the denominator by the former to get a 1 in in the numerator. Now, the operation has become division by a particular number. Since only powers of 2 can perform division in Verilog, we now need to approximate the number in the denominator to a power of 2 number. Once we get that particular number, division can be done by shifting the signal (or delayed signal) right by $\log_2 of\ that\ number$.

The input and output waveforms for each of the filters are shown on the next page.

Table 2. h(n) values for Band Pass filter

n	0	1	2	3	4	5	6	7	8
h[n]	-.00248	.00009	.0043	.00009	.0245	-.00026	891/50000	-.00063	4543/50000

n	9	10	11	12	13	14	15	16
h[n]	-.00013	-.272	-.00082	.633	-.00082	-.272	-.00013	4543/50000

n	17	18	19	20	21	22	23	24
h[n]	-.00063	891/50000	-.00026	.0245	.00009	.0043	.00009	-.00248

VI. OUTPUT WAVEFORMS

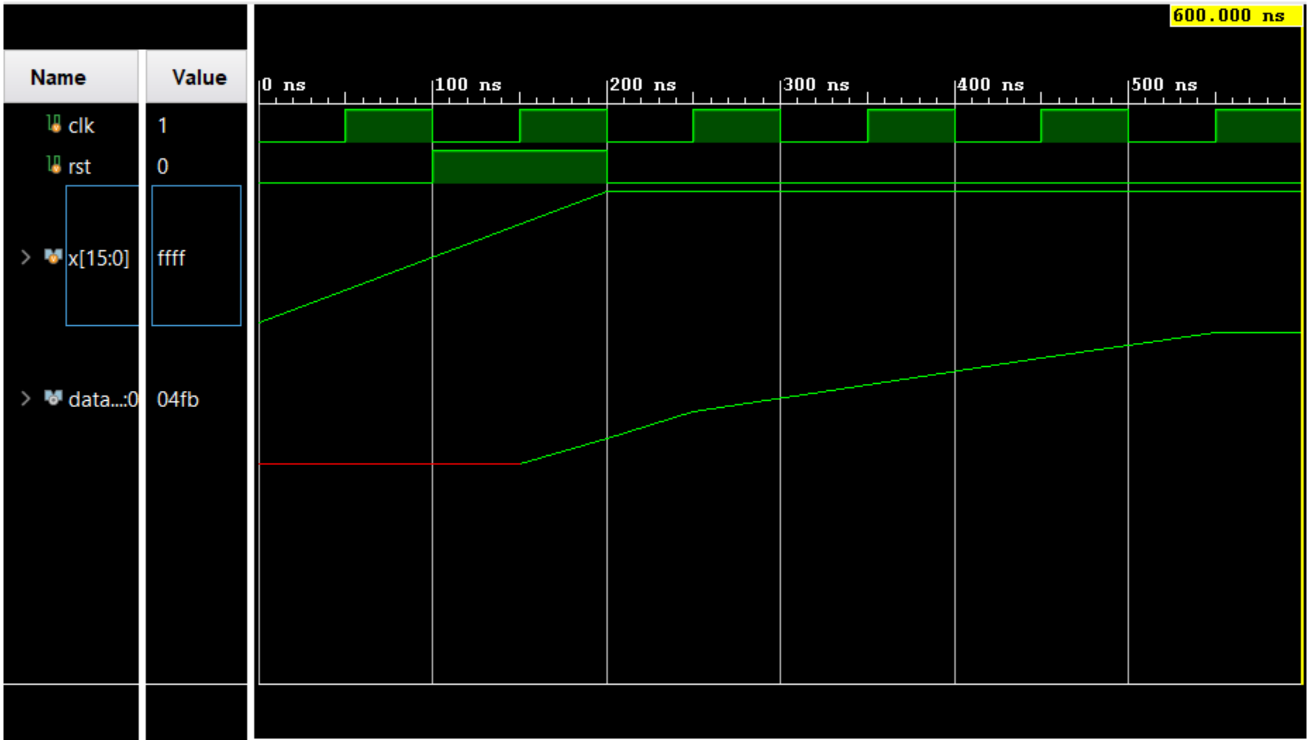

Figure 8 Low Pass Filter Output (Analog)

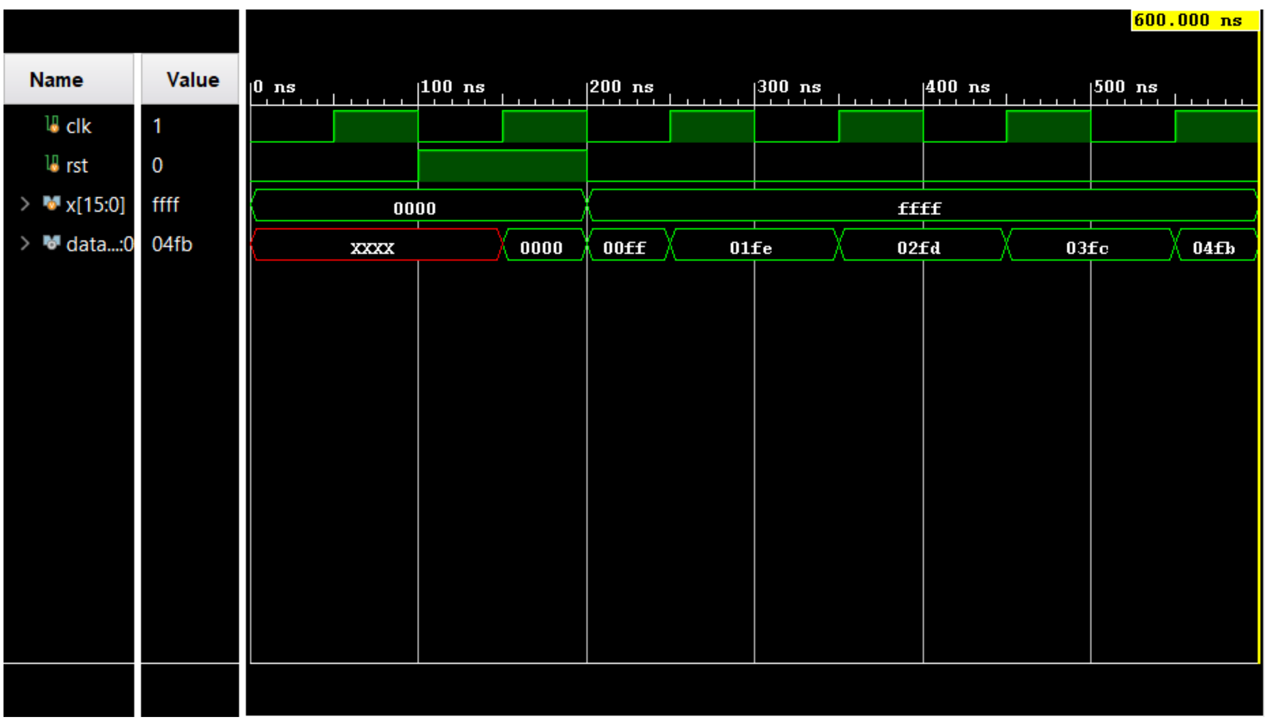

Figure 9 Low Pass Filter Output (Digital)

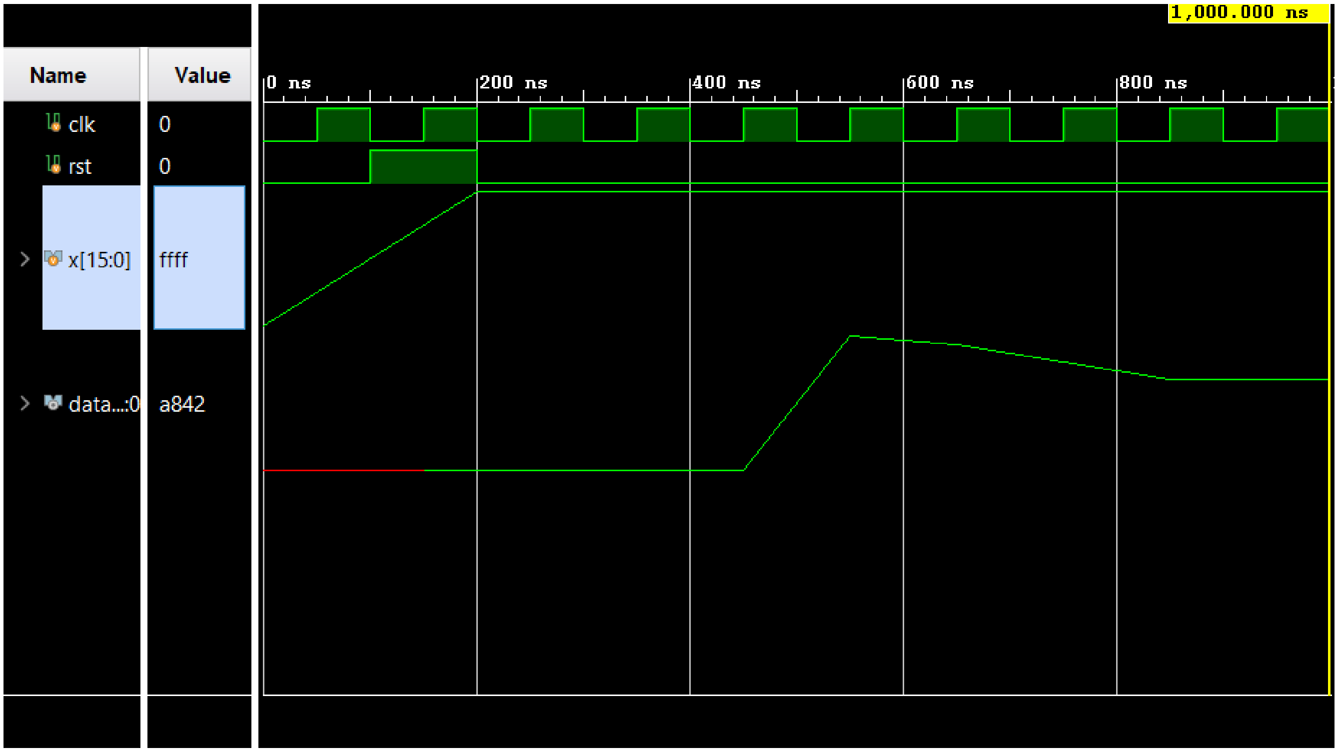

Figure 10 High Pass Filter Output (Analog)

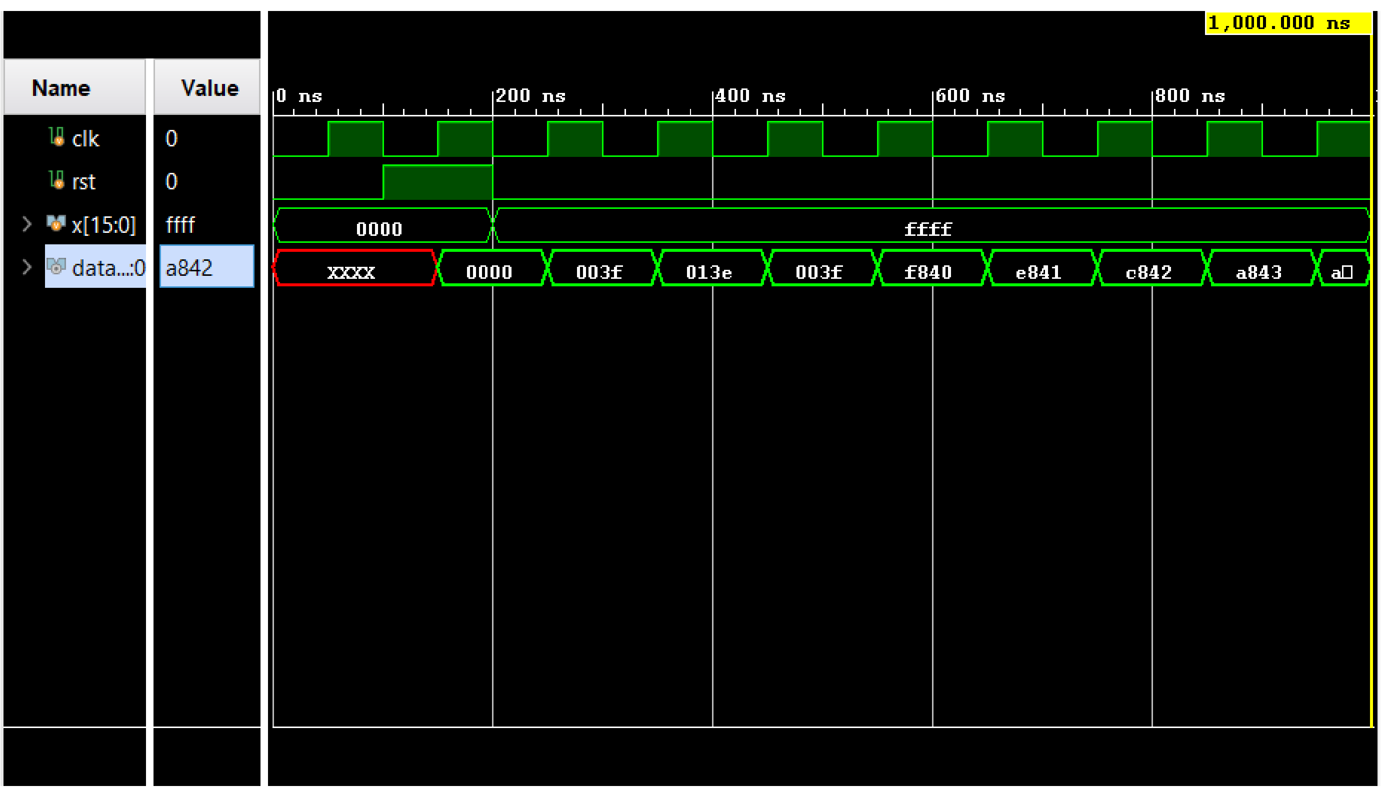

Figure 11 High Pass Filter Output (Digital)

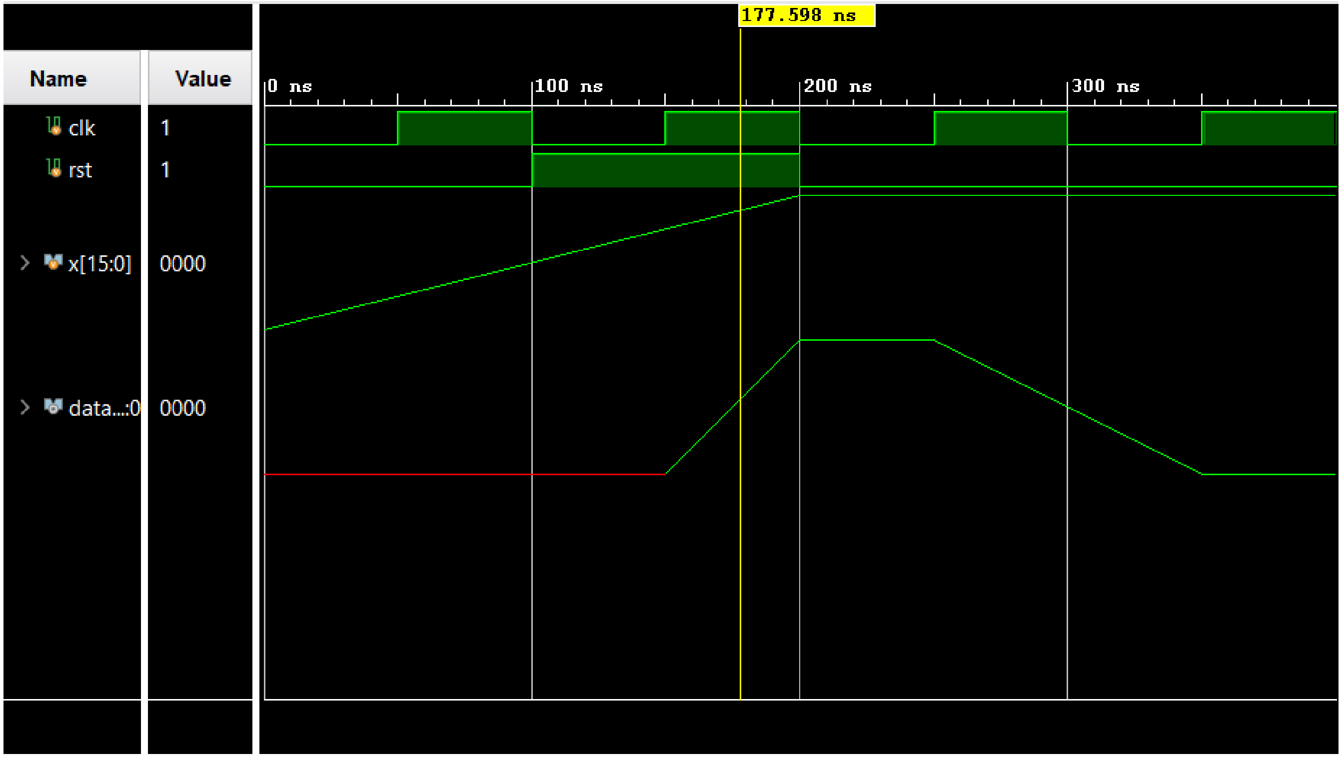

Figure 12 Band Pass Filter Output (Analog)

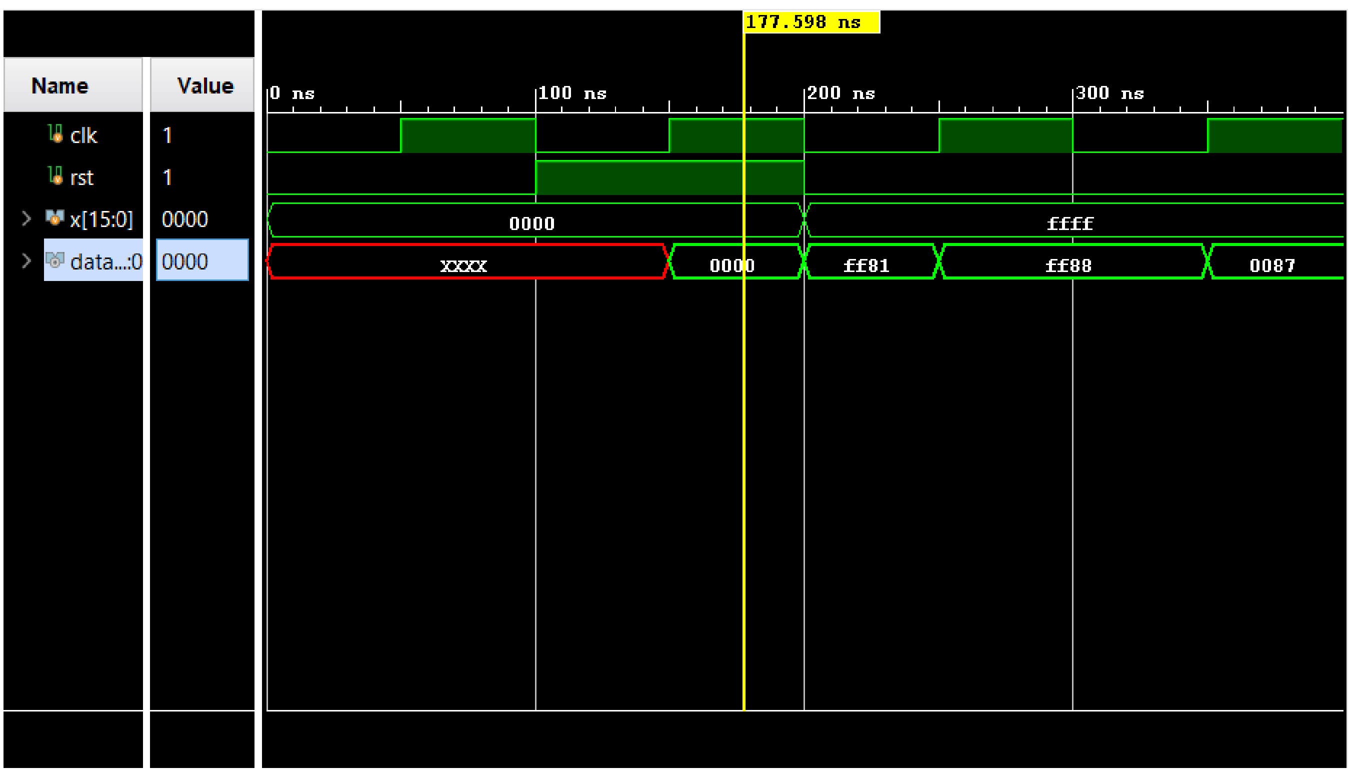

Figure 13 Band Pass Filter Output (Digital)

VII. Acknowledgment

We owe sincere thankfulness to the faculty of Electronics and Communication Branch of Manipal Institute of Technology for providing us with the knowledge with which we were able to do this project work.

VIII. References

[1] Alan V. Oppenheim, Ronald W Schaffer, "Discrete time Signal Processing."

[2] Kolawole, E.S., Ali, W.H., Cofie, P., Fuller, J., Tolliver, C. and Obiomon, P. (2015) Design and Implementation of Low-Pass, High-Pass and Band-Pass Finite Impulse Response (FIR) Filters Using FPGA. Circuits and Systems, 6, 30-48.