Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 12, December-2021

Comparative Analysis of Order Roll off of Filter And Imperative Study of Step Response of Filter using Xilinx IP Core

Vedant Phapale, Siddhesh Chunale, Tanvi Mankapure, Atharva Helskar, Prof. A.V. Chitre.
Department of Electronics and Telecommunication
VIIT Pune, India

Abstract:- **Signal separation and signal restoration are 2 of the most commonly used phrases in digital signal processing. Filtering is an essential part of both their workflows. For removal of undesired components from a signal, a filter is a useful tool in signal processing. Due to their greater adaptability than analogue counterparts, digital filters have becoming increasingly popular and preferred one. In this paper, we'll discuss a low-pass FIR filter constructed in Matlab and its implementation in Xilinx Nexys 4 FPGA. Paper also demonstrated implementation of different fir filter parameters and investigates with filter builder toll. Finally, the paper presents the FIR filter IP core in Vivado platform and Verilog to build the FIR filter designed in Matlab on Xilinx Nexys 4 FPGA.**

I. INTRODUCTION

Using Matlab to design the FIR filter simplifies the intricate computations involved in simulation and improves performance. Matlab processes the design of FIR filter utilizing window function, frequency sampling and convex optimization techniques. The amplitude frequency characterization can be accomplished with the help of Matlab designed application. By comparing the signal's amplitude frequency graphs created by the FIR digital filters used to process the input signal with the Matlab function, the filtering effect of various digital filters is studied. A digital filter is a discrete system that can perform a series of mathematical operations on the input signal in order to extract the needed information from it. The transfer function for a time-invariant liner system.

The experimental results demonstrate the effectiveness of the FIR filters described in their study.

II. FIR FILTER

FIR is an abbreviation for finite impulse response. A FIR filter is the one that has finite period impulse response. The impulse response of a Nth order FIR filter is N+1 before it settles to zero (precisely). A FIR filter is a filter whose impulse response settles to a zero over a specified time period, hence making the response finite. The time required for the impulse response to reach zero is proportional to the order of filter. FIR filter is capable of matching phase and amplitude requirements; it also eliminates draft and noise. The underlying premise of the FIR filter is as follows: The impulse response of a FIR filter h(n) for a series of length N: in addition to meeting phase and amplitude requirements, this filter eliminates drift and noise.

The FIR filter's fundamental premise is as follows: The transfer function of a FIR filter's impulse response h(n) for a series of length N:

$$H(z) = \sum_{n=0}^{N-1} h(n) z^{-n} \tag{1}$$

And the differential equation for the same is:

$$y(n) = \sum_{i=0}^{L-1} x(n-i) h(i) \tag{2}$$

III. FIR FILTER DESIGN:

The Window approach, such as the Kaiser window method, is used to create FIR filters. To construct FIR filters using the window technique, the fundamental principle is to calculate $h_d(n)$ using the anti-Fourier transform based on the ideal desired filter frequency response. $H_d(ejw)$:

$$h_d(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_d(e^{jw}) e^{jwn} \, dw \tag{3}$$

As the formula for $h_d(n)$ is infinitely long, to get the unit impulse response h(n) we have to use the window function to meet the requirement. The

formula is:

$$h(n) = w(n) \bullet h_d(n) \tag{4}$$

w(n) is the window function in this case. Filters are the devices that allow only some signals to pass through while rejecting the rest. It is necessary to apply the same idea to LIA (lock in amplifier), which is why the use of a filter is essential.

IV. DESIGN OF FIR FILTER

The overall methodology of experimentation is demonstrated in the following figure:

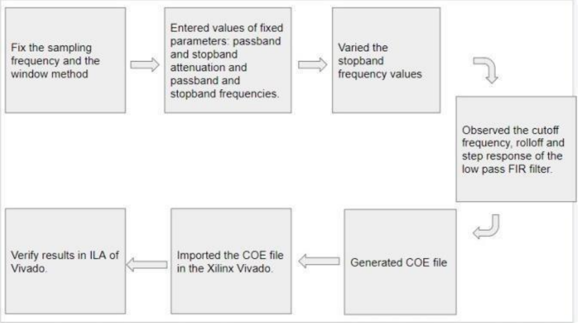

Fix the sampling frequency and the window method
Entered values of fixed parameters: passband and stopband attenuation and passband and stopband frequencies.
Varied the stopband frequency values
Observed the cutoff frequency, rolloff and step response of the low pass FIR filter.
Generated COE file
Imported the COE file in the Xilinx Vivado.
Verify results in ILA of Vivado.

Fig 1.1 Block diagram of the experiment

The magnitude characteristic of the frequency response is determined by FIR filter with a polynomial transfer function in the z axis and zeroes in the z- plane. FIR filters produce a phase response that is perfectly linear. A non-recursive strategy is used to construct the FIR filter, which results in a stable filter. Two sections comprise the FIR filter design

1) An preliminary analysis of the given problem
2) In-depth analysis of the problem
3) In the analysis step, we take the specified object and generate a transfer function.

The next step to specify the transfer function:

I. An optimal response is chosen, which is often in the frequency domain.
II. For FIR filters, the length M is chosen.
III. A criterion for the analysis's quality is chosen.
IV. To determine the optimal filter transfer function, a flowchart is used.

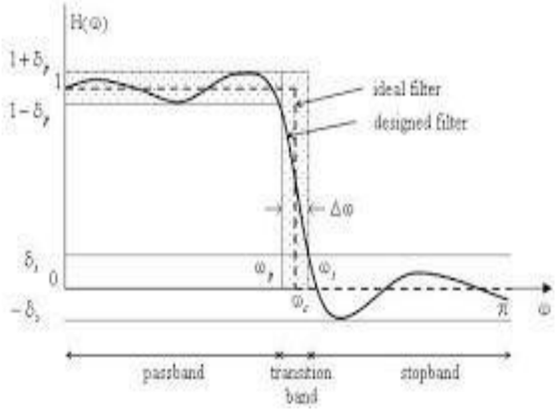
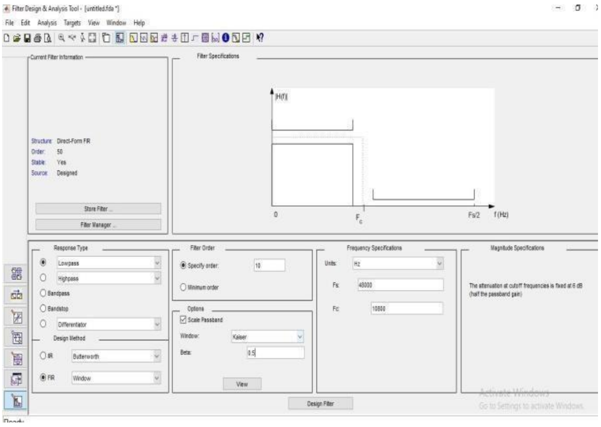


Fig. 1.2 Specification of LPF filter

As illustrated above, the transition band is always located between the pass-band and the stop-band of filter. The pass-band frequency s determines the stop- band frequency p, and the pass-band frequency p marks the pass-band edge. Thus, transition band's width is determined by the difference between ωs and ωp, δs ripple or attenuation in the stop-band. The ripple or attenuation in the pass-band of the filter is denoted as δp, and the magnitude of the filter varies from -δp to δp. The ripple in the pass-band is 20log10δp dB, and the ripple in the stop-band is 20log10δs dB.

Mat-lab provided FDA tool automatic set the ripple at cutoff frequencies fixed at 6 db (half of pass-band gain)



Structure: Direct-Form FIR
Order: 50
Stable: Yes
Source: Designed

● Lowpass
○ Highpass
○ Bandpass
○ Bandstop
○ Differentiator

○ IIR Butterworth
● FIR Window

● Specify order: 10
○ Minimum order

☑ Scale Passband
Window: Kaiser
Beta: 0.5

Units: Hz
Fs: 48000
Fc: 10800

The attenuation at cutoff frequencies is fixed at 6 dB (half the passband gain)

Fig 1.3 FDA Tool In Matlab

V. WINDOW FUNCTION:

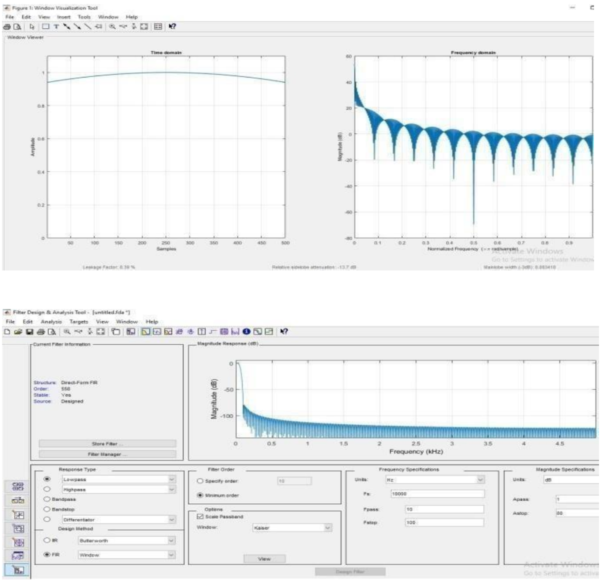
In many cases of stable fir filter most selective approach to obtain a causal finite impulse response is to truncated or smoothen the ideal unit response.

We start with fixed (desirable) frequency response in filter specification H d(e ^jw) and determine unit sample response (n), hence H d(n) is correlated to h d(e^ jw) by Fourier transform relation.

Some example of window function can be used practically

1) Triangular window
2) Hann window
3) Hamming window
4) Kaiser window

In our project we use Kaiser window function because we want stable filter.



Leakage Factor: 0.39 %
Relative sidelobe attenuation: -13.7 dB
Mainlobe width (-3dB): 0.003418

Structure: Direct-Form FIR
Order: 558
Stable: Yes
Source: Designed

● Lowpass
○ Highpass
○ Bandpass
○ Bandstop
○ Differentiator

○ IIR Butterworth
● FIR Window

○ Specify order: 10
● Minimum order

☑ Scale Passband
Window: Kaiser

Units: Hz
Fs: 10000
Fpass: 10
Fstop: 100

Units: dB
Apass: 1
Astop: 80

Fig 2.1 Magnitude and Frequency response

We understand window function method very well with the help of Matlab FDA TOOL

Below is the graph of order of filter vs cut off frequency:

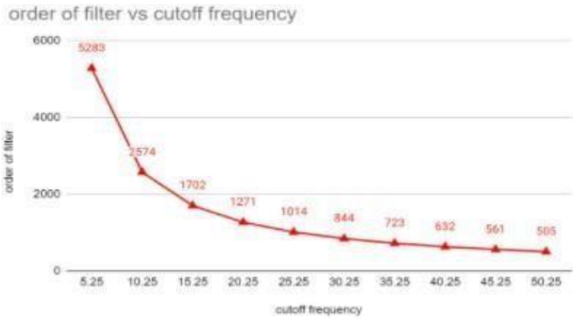


Fig 3.1order vs cutoff frequency

Here, the less the difference between fpass and fstop the order of filter will be high and vice versa. This graph is obtained by using the filter function from matlab. One can obtain a similar graph, for that keep the fpass value constant. And then vary the fstop. In our case, we have varied the fstop from value: 900 to 1000 hz. By varying these values we observed that the difference between fpass and fstop started increasing and so the value of order of filter started dropping down.

Below is the graph of order of filter vs settling time:

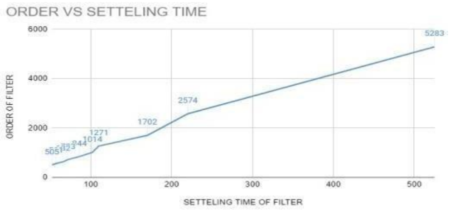


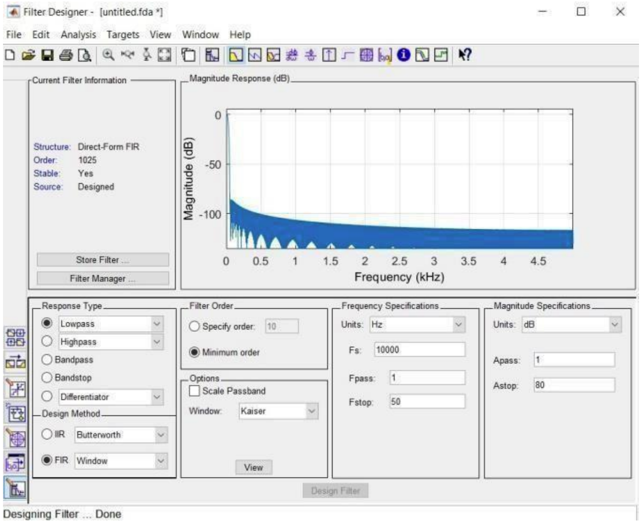
Fig 3.2 order of filter vs settling time

The above graph is obtain by using the filter function from Matlab. One can obtain a similar graph, for that keep the fpass value constant and vary fstop. In our case we have varied the fstop from value 10 to 100 hz. By varying these value we observed that difference between fstop and fpass so the value of order of filter start decreasing and we obtain sattling time using step impulse function we analysis that for maximum order of filter value maximum settling time.

VI. KAISER WINDOW:

Kaiser window parameter are affecting the sidelobe and attenuation of fourier transform of the window in our case it's value by default for beta is 0.5. To verify a Kaiser window design in FIR Filter with sidelobe attenuation of α db use following β Syntax in matlab y=kaiser(k,β)

Where y=kaiser(k,β) return an k- point Kaiser window in the column vector y And β is Kaiser window parameter **Example y=kaiser(500,0.5)**


Filter Designer - [untitled.fda *]

Current Filter Information

Structure: Direct-Form FIR
Order: 1025
Stable: Yes
Source: Designed

Response Type
● Lowpass
○ Highpass
○ Bandpass
○ Bandstop
○ Differentiator

Design Method
○ IIR Butterworth
● FIR Window

Filter Order
○ Specify order: 10
● Minimum order

Options
□ Scale Passband
Window: Kaiser

Frequency Specifications
Units: Hz
Fs: 10000
Fpass: 1
Fstop: 50

Magnitude Specifications
Units: dB
Apass: 1
Astop: 80

Designing Filter ... Done

Fig 3.3 Frequency Response of FIR Filter order 1025

Here, we can observe that for certain parameter we obtain 1025 order which is more essential for stable filter design.

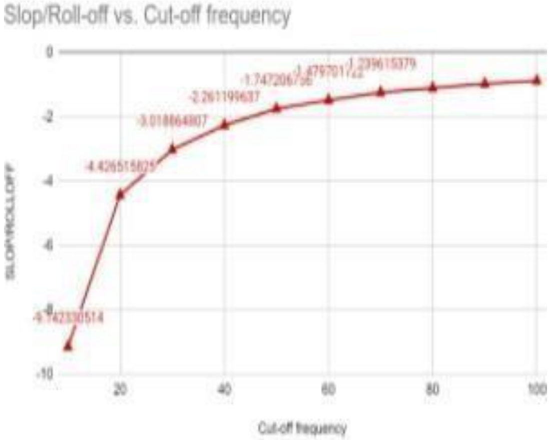


Fig 3.4 rolloff vs cutoff frequency

We also observed the behavior of roll off vs cutoff of a low pass FIR filter. The graph above is the final result. Here, roll off is the steepness observed in a transfer function with certain frequency value.

The graph was obtained on Matlab's filter function, where the value of cutoff frequency was decreased. By doing so it was observed that the slop or the roll off started becoming steeper.

Here's the roll off gets steeper as the values of Fpass and Fstop are close to each other. On the other hand, as the difference between Fpass and Fstop is small the rolloff isn't steep. For a gradually changing roll off the result will be a lower flter order.

VII. NEXYS 4 BOARD ARTIX 7 FPGA DDA BOARD:

The Nexys4 board is a complete development platform based on the latest Artix-7 FPGA. The Nexys4 BOARD is compatible with Xilinx's optimized for high performance logic, and offers more capacity, higher performance, and more resources than earlier designs.

In this project we required 15879 logical dsp slices each having 6 LUTS AND 8 flip flop also BRAM is required 4800 kbits and 240 DSP slices is used for reduce dsp utilization

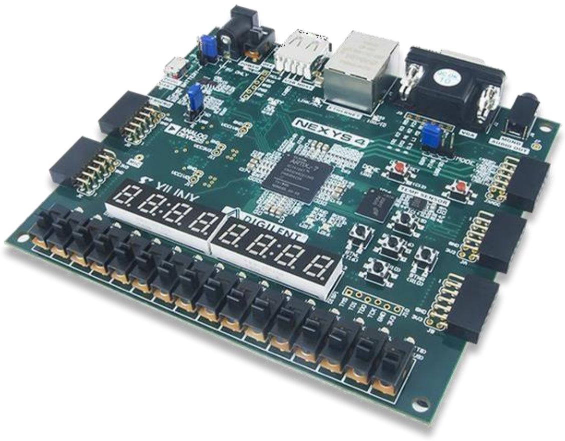
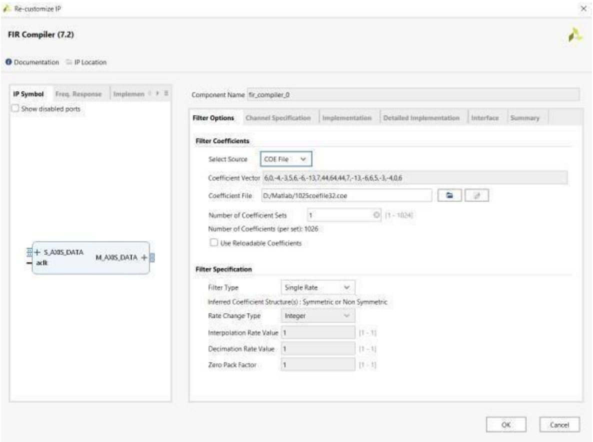


Fig 4.1 NEXYS 4 BOARD ARTIX 7 FPGA

VIII. FPGA IMPLEMENTATION:

For trying out the fir filter built in the Filter designer tool of mat-lab, we created a coefficient file from mat- lab and imported it into the Fir ip core in Xilinx Vivado. After interesting the coefficients file we integrated the ip into our verilog code. We also used integrated logic analyzer ip of xilinx to observe the simulation results of our code. The code is written in verilog HDL.



Re-customize IP

FIR Compiler (7.2)

Component Name fir_compiler_0

Filter Coefficients

Select Source: COE File

Coefficient Vector: 6,0,-4,-3,5,6,-6,-13,7,44,64,44,7,-13,-6,6,5,-3,-4,0,6

Coefficient File: D:/Matlab/1025coefile32.coe

Number of Coefficient Sets: 1 [1 - 1024]

Number of Coefficients (per set): 1026

☐ Use Reloadable Coefficients

Filter Specification

Filter Type: Single Rate

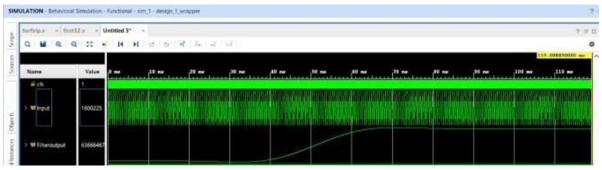
Inferred Coefficient Structure(s) : Symmetric or Non Symmetric

Rate Change Type: Integer

Interpolation Rate Value: 1 [1 - 1]

Decimation Rate Value: 1 [1 - 1]

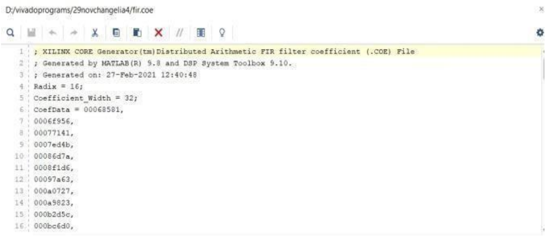
Zero Pack Factor: 1 [1 - 1]



SIMULATION - Behavioral Simulation - Functional - sim_1 - design_1_wrapper

Name	Value
clk	1
Input	1600225
Filteroutput	63666467

Fig 5.1 FPGA implementation using Verilog code

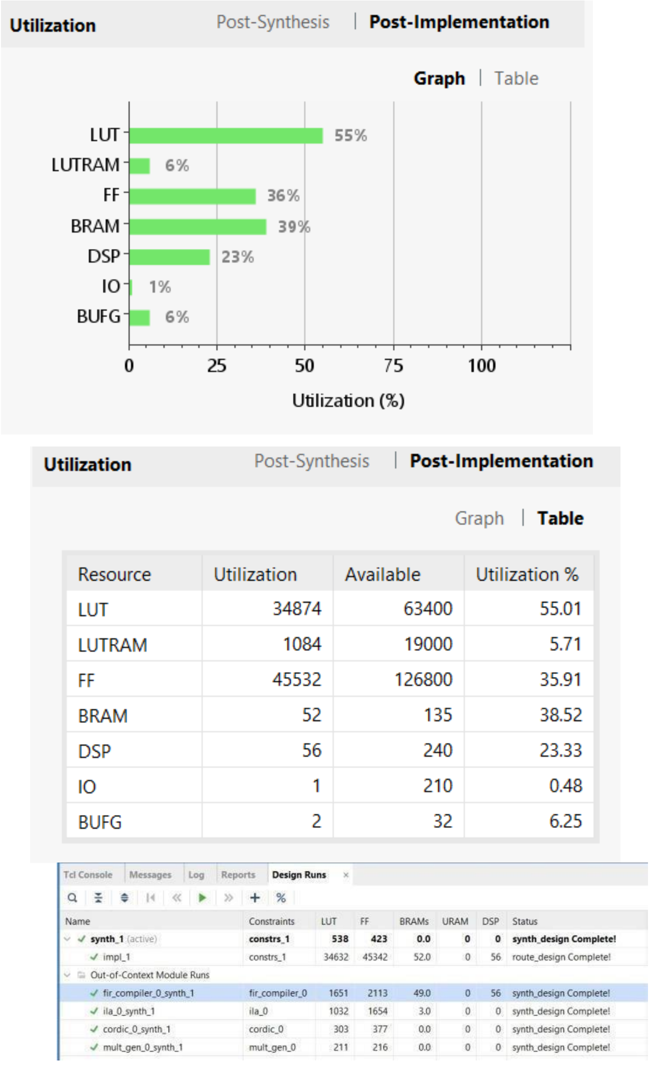


D:/vivadoprograms/29novchangelia4/fir.coe

```
; XILINX CORE Generator(tm)Distributed Arithmetic FIR filter coefficient (.COE) File
; Generated by MATLAB(R) 9.8 and DSP System Toolbox 9.10.
; Generated on: 27-Feb-2021 12:40:48
Radix = 16;
Coefficient_Width = 32;
CoefData = 00068581,
0006f956,
00077141,
0007ed4b,
00086d7a,
0008f1d6,
00097a63,
000a0727,
000a9823,
000b2d5c,
000bc6d0,
```

Fig 5.2 Coefficient file as seen in Xilinx

Utilization report:



Utilization | Post-Synthesis | Post-Implementation | Graph | Table

Resource	Utilization	Available	Utilization %
LUT	34874	63400	55.01
LUTRAM	1084	19000	5.71
FF	45532	126800	35.91
BRAM	52	135	38.52
DSP	56	240	23.33
IO	1	210	0.48
BUFG	2	32	6.25

Tcl Console | Messages | Log | Reports | Design Runs

Name	Constraints	LUT	FF	BRAMs	URAM	DSP	Status
✓ **synth_1** (active)	**constrs_1**	**538**	**423**	**0.0**	**0**	**0**	**synth_design Complete!**
✓ impl_1	constrs_1	34632	45342	52.0	0	56	route_design Complete!
Out-of-Context Module Runs							
✓ fir_compiler_0_synth_1	fir_compiler_0	1651	2113	49.0	0	56	synth_design Complete!
✓ ila_0_synth_1	ila_0	1032	1654	3.0	0	0	synth_design Complete!
✓ cordic_0_synth_1	cordic_0	303	377	0.0	0	0	synth_design Complete!
✓ mult_gen_0_synth_1	mult_gen_0	211	216	0.0	0	0	synth_design Complete!

Fig 5.3 FPGA resource utilization

We saw synthesis report and utilization report generated by the Vivado platform where we clearly see the Filter fir ip uses 56 DSP cells out of 240 DSP cells present in the nexys 4 FPGA board.

IX. FUTURE SCOPE:

Digital filters give us advantages when compared to analog filters, in terms of reprogrammability and repeatability. Also, the digital filters are less sensitive to environmental changes, noises and disturbances. These features of digital filter can be used in several applications. Further, we will be use this optimum filter design in digital lock in amplifier on an FPGA board.

X. CONCLUSION:

1) FIR filter design generally consists of two sections.
2) Analysis of problem
3) Realization problem with deep approach
4) Basically, there are three approaches for FIR filter design:
5) Function window approach
6) frequency sampling technique
7) Optimal filter design approach
8) The major advantages of using window method is their relative simplicity as compared to other methods and ease of use.

XI. REFERENCES:

[1] Practical FIR Filter Design in MATLAB, Ricardo Losada, The MathWorks, Inc, 3 Apple Hill Dr. Natick, MA 01760, USA January 12, 2004.

[2] Prabhat Ranjan "Implementation of FIR Filter on FPGA" Department of Electronics and Communication Thapar University, Punjab (July 2008).

[3] Design and verification of FIR filter based on MATLAB and DSP, Chen-Long Hu, IEEE 2012 International Conference on Image Analysis and Signal Processing (IASP), 10.1109/IASP.2012.6425042

[4] http://www.mathworks.com/help/toolbox/signal/fd atool.html

[5] Nexys 4™ FPGA Board Reference Manual Revised April 11, 20