# Hardware Implementation of IIR Digital Narrow Band Stop Filter

**Kishan Kumar Das, Vikas Pathak, Neeraj Jain**

Department of Electronics and Communication Engineering, Swami Keshvanand Institute of Technology, Management and Gramothan,Jaipur, India

***Email:*** **kishankumard535@gmail.com, vikas.pathak@skit.ac.in, neeraj.jain@skit.ac.in**

**Received** 01.09.2023 **received in revised form** 30.03.2023, **accepted** 16.10.2023

**DOI: 10.47904/IJSKIT.13.2.2023.23-29**

***Abstract*— Digital filter is a system, which applies a mathematical operation to discrete time signal x[n] in order to diminish or increase the particular aspect of signal. In this manuscript, an infinite impulse response (IIR) digital Narrow Band Stop (NBS or Notch) filter is employed. In filter design, a band stop filter, also known as a band reject filter, is a filter that allows most frequencies to be unchanged but has a very low level in a particular range. A band-stop filter with a condensed stopband is a notch filter or Narrow Band Stop (NBS) filter. This paper carries out the hardware implementation of the digital Notch (NBS) filter. We have used IIR direct form- II structures for implementation. Time-changing bandwidth notch filters have been implemented using IIR structures. Xilinx ISE Tool is used to implement the suggested design on an FPGA. This Notch filter is also validated using MATLAB for sampling frequency of 1000 samples/second with notch frequency of 300 Hz. Here the input test data is taken with three frequency components of 100, 300 and 700 Hz. The hardware and software (MATLAB) implemented results clearly stop the 300 Hz frequency component with improved timing.**

***Keywords*— *Hardware implementation, Notch filter, FPGA, IIR filter, Direct form structure.***

## I. INTRODUCTION

The Notch filter, which is used to eliminate a single frequency or a narrow frequency, has a wide range of applications. When it comes to the audio system, a notch filter can be utilized to eliminate disruptive frequencies like power-line hum [1]. Additionally, it can be used to disable a particular set of interfering frequencies in software-defined radio and radio receivers.

The filtering activity that needs to be carried out on a continuous time signal is implemented by a digital filter using computation. A block diagram of the process used in this method of frequency-selective filter design is displayed in figure 1. The continuous-time signal x (t) is converted into a corresponding sequence of numbers using the "Analog-to-digital (A/D) converter" block [2]. The digital filter transforms the sequence of numbers x[n] into a new sequence of numbers y[n], which is then translated into the corresponding continuous time signal by the digital-to-analog (D/A) converter on a sample-by-sample basis by the digital filter. At the system's output, the reconstruction (low-pass) filter [3] creates a continuous-time signal y (t), which represents the filtered form of the original input signal x (t).

When researching digital filters, it's crucial to keep the following two things in mind: To take advantage of well-understood discrete time, the fundamental design method is typically based on the usage of analogue or infinite precision models for the sample of the input data and all internal calculations.

The input data and its internal operations are all quantized to a finite precision time, whenever another discrete time filter [4] is built in a digital form for actual usage, as we described in figure (1). Whenever we consider that the discrete time filter [4] is implemented in a digital form for practical use, then as we mentioned in the figure (1), the input data and the internal manipulation are all quantized to a finite precision time.

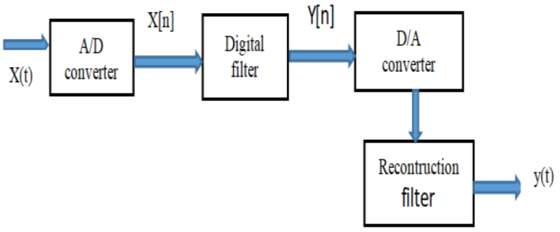


**Figure-1:** A continuous-time signal filtering system, made for a digital filter [2].

Using either the finite impulse response (FIR) order form or the infinite impulse response (IIR), the digital notch filter can be created. An IIR notch filter is preferred over a FIR notch filter for reducing the computational complexity and signal technique delay. But the IIR notch filter have a transient response and might damage the starting portion of processed signal. Filter order is responsible for transient response duration, so filter designer uses the lower order notch filter [5]. Dr. Joseph H. Piskorowski proposed an answer in an effort to suppress the initial time varying analysis with notch bandwidth [6].

A function that varied in time was used to gradually expand the pole radius to the desired amount, starting with a tiny value for the pole radius. When a result, the initial samples had a broad notch width, and as the transient diminished, the notch width shrank. In [7], an enhanced time-varying pole radius IIR notch filter was suggested. The time-dependent change in the pole radius is represented by an exponential function with a damping parameter. The transient duration can be predicted using a method that was proposed using sinusoidal interference analysis. The damping factor was then calculated using the projected duration. The hyperbolic tangent sigmoid (HTS) function was then used by Rana et al. [8, 9] to produce time-varying pole radius coefficients. Once the efficient transient suppression is attained, the HTS function permits the pole radius to fast change from minimum to maximum values. The typical IIR notch filter can be used to deduce the implementation structure of the notch filter with time-varying notch width.

In the given novel [10] the novelist tells about the pole radius IIR notch filter while enhancing the performance. This study compares and contrasts various approaches to changing an infinite impulse response (IIR) digital notch filter's pole radius. To prevent sinusoidal interference in Electrocardiogram (ECG) signals, it is important to have a short transient length and a high-quality factor. MATLAB Simulink and the Xilinx System Generator for DSP are used to implement the suggested notch filter.

A second-order time-varying pole radius IIR notch filter is presented in this study. The pole radius is changed over time to shorten the transient duration in the filter output. In this work, three distinct functions exponential, hyperbolic sigmoid, and gamma are provided to produce the variable pole radius. The IIR notch is implemented with the help of the direct form-II structure. By comparing the root mean square error values for the various functions, the performance of the notch filter is assessed. A variety of fixed-point representations are also used to create the filter structure on the Xilinx System Generator. The number of Look-Up Tables (LUTs) and registers needed for FPGA implementation are estimated using the resource analysis tool.

In paper [11] the novelist tells about the transient duration lattice wave digital notch filter with FPGA implementation. In this work, a digital notch filter's design and FPGA implementation using the lattice wave digital filter (LWDF) structure are presented. By creating a variable notch bandwidth filter, the initial signal transient is intended to be reduced. The notch filter's wide bandwidth is initially used to reduce the signal transient. A notch filter with a low transient duration and high-quality factor is produced by gradually reducing the notch bandwidth to attain the narrowest width. The Infinite Impulse Response (IIR) structure was previously used to create time-varying bandwidth notch filters, and this structure needed two variable coefficients to change the notch width over time. Utilizing the LWDF structure has the benefit of requiring only one coefficient to change in order to reach the desired notch width with oscillation over time.

The amount of memory needed for implementation is halved as a result. The LWDF is also more robust since word-length effects are less likely to affect it. As a result, when compared to previous research, the suggested lattice wave digital notch filter (LWDNF) produces better results for error analysis. Utilizing the Xilinx System Generator for DSP design suite, the proposed LWDNF architecture is implemented on an FPGA.

The hardware implementation of IIR narrow band stop filter can be used in many applications fields such as DNA sequence coding [12, 13]. Due to which DNA in each chromosome can be read as a discrete signal. On the other hand, Digital filters can be used for Exon localization in DNA sequences. DNA sequence is numerically mapped into four binary indicator sequences. Each indicator sequence is filtered to produce the filtered output sequence using the proposed narrow band stop filter.

Using Android devices, users can apply Effects to audio signals utilizing android devices and deploy band-pass filters using ‘Android-powered’ gadgets for deploying Biquad and IIR notch filters for noise reduction.

In this manuscript, hardware realization of IIR notch filter is carried out. For this IIR direct form second structures are used due to its lesser requirement of hardware units like adder and multiplier. Further to reduce the hardware only second order IIR structures are implemented.

To design the second order structure of IIR filter, it is taken into account the hardware implementation of FPGA system, IIR filter, Direct 2nd form structure. Additionally considered here are the notch filter's magnitude responses, poles, and zeroes, as well as the transfer function T (z) IIR all pass filter, demonstrating the notch filter's robust design.

## II. IIR NOTCH FILTER DESIGN

FIR filter, which has such responses will be required for long impulses responses whenever it is implying more computations. While we look for IIR filter it will require much shorter order which can be very efficient for analysis. Such filter which can be made up from second order all pass filter. That’s why IIR filter is selected for notch filter execution.

Therefore, second order all-pass filter at with real pole coefficient $re^{\pm j\alpha}$ [14, 15]. So that the second order IIR all pass filter's transfer function may be expressed as.

$$Q\ (Z) = \frac{r^2 - 2\,r\cos(\alpha)Z^{-1} \ + Z^{-2}}{1{-}2r\cos(\alpha)Z^{-1} \ +r^2\,Z^{-2}} \qquad (1)$$

Here the frequency of notch is $\alpha$ and radius of the pole is ‘$r$’. As we know that for causality of LTI system whenever $r^2 < 0$ for stability condition. Although $r$ is satisfying the condition of Bounded Input and Bounded Output (BIBO). As we noticed the above equation the numerator part is the symmetric of denominator part so that, the zeros are at the reciprocal location $\frac{1}{re^{\pm j\alpha}}$ . Now, we can assume the P (z) towards the pole’s sides, which is given as Q $(e^{j\omega})$ $=\frac{e^{-2j\omega}\,P*(e^{j\omega})}{P(e^{j\omega})}$. This condition satisfied the all-pass filter property |Q $(e^{j\omega})$| = 1.

Now, we can have considered the notch filter form equation as,

$$T\ (z) = \frac{1+\ Q(Z)}{2} \qquad (2)$$

Therefore, the generalized form of transfer function of Notch filter equation (derived from equations 1 & 2) is given as follows

$$T(z) = k\left(\frac{1-2\cos(\beta)Z^{-1}\ +Z^{-2}}{1{-}2r\cos(\alpha)Z^{-1}\ +r^2Z^{-2}}\right) \qquad (3)$$

Therefore, k = (1 $+ r^2$ ) / 2 and $\cos\beta = 2r\cos\alpha$ / $(1+\ r^2)$. Thus, we can say that T (z) is the zeros which can be depicted on the unit circle makes an angle of ‘ $\beta$ ’. As we can say that the pole is approaching to the unit circle( r → 1), due to which $\alpha$ is roughly equal to $\beta$ ($\alpha \approx\ \beta$ ). Hence, the filter T (z) has zeros and poles more too closer each other. Therefore, the small change in frequency response as well as magnitude response which is away from the angle of $\beta$. Such filter is known as notch filter.

The IIR filter is recursive in the sense that it computes its output using its inputs, outputs, and both the current and prior inputs. As a result of the filter using earlier output values, the filter structure incorporates feedback on the output. [16].

Substitute the value k = (1+ $r^2$) /2 and $\cos\beta = 2r\cos\alpha\,/\,(1+\ r^2)$ or 2r cos $\alpha$ = $(1+ r^2)$ $\cos\beta$ in the equation (3), we get the final transfer function of Notch filter as given below

$$T(z) = \frac{1}{2}\left(\frac{1{+}\,r^2{-}\,4r\cos(\propto)Z^{-1}\ +1{+}\,r^2Z^{-2}}{1{-}2r\cos(\alpha)Z^{-1}\ +r^2Z^{-2}}\right) \qquad (4)$$

The transfer function for the generalized standard form of the IIR filter is as follows:

$$\frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} +\ b_2 z^{-2} \ldots\ldots\ldots\ldots b_n z^{-n}}{1 + a_1 z^{-1} +\ a_2 z^{-2} \ldots\ldots\ldots\ldots a_m z^{-m}} \quad (5)$$

Therefore, the reverse coefficient is $b_n$ and $a_m$ is the forward coefficient of the filter.

The coefficient of the IIR notch filter can therefore be determined by comparing equations (4) and (5).

$$\begin{bmatrix} b \\ a \end{bmatrix} = \begin{bmatrix} \frac{1+r^2}{2} & -(1+r^2)\cos\beta & \frac{1+r^2}{2} \\ 1 & -(1+r^2)\cos\beta & r^2 \end{bmatrix} \qquad (6)$$

Therefore radius ‘r’ at pole and ‘β’ is the frequency of notch filter. The zeros are positioned in this operation at the angle that corresponds to +β at the z unit circle, we can assume that when Z = 1 and β = 0 for lower frequency and Z = -1 and β = $\pi$ for higher one. The poles are situated inside the close area of zeros' unit circle due to which angular frequencies β0 has unit gain (as indicated in figure 3). The notch of the filter which is observed by specific value r (0.9 << r < 1). In filter design technique a band stop filter, also known as a band reject filter, allows most frequencies to pass through unaltered but has an extremely low level in a particular range. Band-stop filters with a small stopband are known as notches.

- In this paper we have taken the IIR second order of filter.
- Filter type is IIR Narrow band reject filter (NF) which reject a particular frequency whose coefficient is taken from all pass filter with pole radius r
- Poles radius is taken as r (0.9 < r < 1), typical value chosen is 0.992

Thus, figure 2 shows the notch filter’s output frequency response as result.

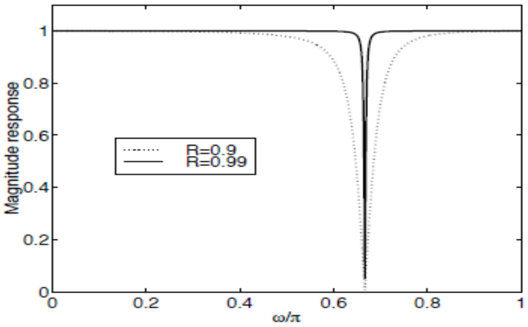


**Figure-2**: Magnitude response at R

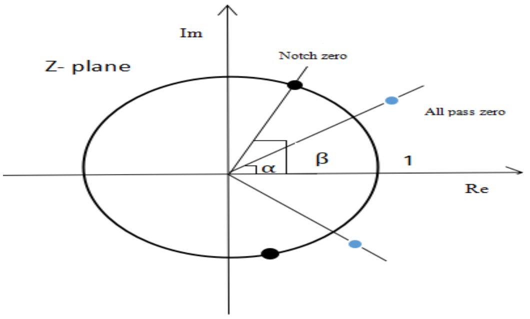


**Figure- 3:** Unit circle view of transfer function T (z).

## III. IMPLEMENTATION OF NOTCH IIR FILTER

The hardware implementation of IIR notch filter is covered in this section. Notch filter can be implemented by both IIR and FIR structure form, but in this paper, we have used Notch filter IIR (Infinite Impulse Responses) with direct second form structure. If we talk about the comparison of IIR filter and FIR, then it implies that IIR filter are well suited for application area, and it will not be required for phase information. For this reason, we are utilizing an IIR filter rather than a FIR filter.

Infinite impulse response filter is recursive filter, and which is relation between input and output. Since its response is Zero, the infinite impulse response filter cannot reach the Zero.

From hardware implementation point of view, it is possible to implement the IIR notch filter using either the direct form-I or direct form-II structure (according to equation 6) described in given below subsections.

### A. *IIR FILTER WITH DIRECT FORM 1st STRUCTURE*

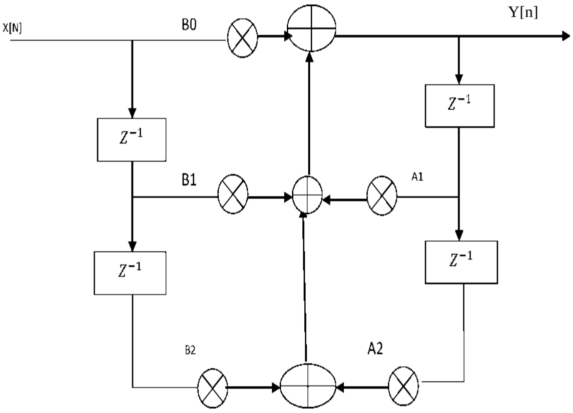


**Figure-4:** Direct form-I structures for IIR notch filter

A simple method where the difference equation is immediately evaluated is the IIR direct form-I structure (as shown in figure 4).

For a filter of order N, this form needs 2N delay elements for both the input and output signals.

### B. *DIRECT FORM 2nd STRUCTURE IIR FILTER*

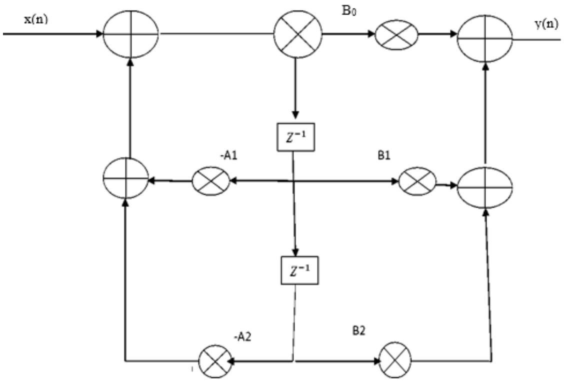


**Figure-5:** IIR notch filter with direct-II form structure [7]

- Direct Form 2nd structure (as shown in figure 5) have N delay for the N order of the filter.
- Since the denominator and numerator of the direct form 1st are in a linear system, we can apply the commutative principle to generate the direct form 2nd from the direct form 1st.

Due to the need for a smaller number of delay components, direct form-II structure is used in this paper (flip-flops). Here, the filter's coefficients and input data are applied which is real data type. So, the data flow here will be of real data. To represent the real data in binary form two options are their fixed point and floating-point number system. Due to its greater precision and range when compared to fixed-point, floating-point arithmetic is recommended in this situation.

In this paper, we are studying the IEEE 754 standard for 32-bit single precision floating point [16]. So as required in figure 4 and 5 floating point adders and multiplier blocks are designed and developed [17, 18]. Then these adders and multiplier units are combined according to figure 5, to make the final IIR narrow band stop (notch) filter.

## IV. RESULT AND DISCUSSION

We have proposed various aspects of notch filter with help of following tools as shown in below.

### A. *MATLAB TOOL*

The math work company created the high-level programming language and environment known as MATLAB. In fact, MATLAB is an effective tool for a variety of technical and numerical computing jobs. Based on the data you gave; the following are some important points and abilities. Numerous paradigms are supported by MATLAB, such as symbolic computing, numerical computing, and several programming paradigms. Through the use of MATLAB, users can design graphical user interfaces (GUIs) for their programmers, simplifying the interaction and visualization of data. The programming and visualizations tools in MATLAB are tightly interwoven, enabling users to design insightful plots and charts. We're working using MATLAB R2014a. It's significant to note that MATLAB has undergone multiple iterations since that time, with each one bringing new capabilities and enhancements.

### *B. XILINX ISE 14.4 TOOL*

A popular tool for creating digital circuits that are intended for Xilinx Field Programmable Gate Arrays (FPGAs) is Xilinx ISE (Integrated Synthesis Environment). Using Xilinx ISE, creating digital circuits often entails the following crucial steps:

- Project Creation: In Xilinx ISE, the first step is to create a brand-new project. A project is essentially a storage space for all the design documents, restrictions, and configuration options pertaining to its digital circuit.
- Design Entry: A procedure for inputting our digital circuit design is covered in this stage.
- Simulations: Running simulations of the design before synthesizing it is a good idea to ensure its accuracy and utility.
- Synthesis: After the concept has been simulated and set up, we proceed. A gate-level netlist, which represents the actual logic gates and flip-flops that will be implemented on the FPGA.
- Implementation: Following the synthesis stage, then go on to the implementation phase. In order to do this, the synthesized netlist must be mapped to the precise resources on the target FPGA device.
- Programming of FPGA: The FPGA must then be programmed using the produced bit stream in the last phase.

### *C. FLOW CHART OF NOTCH FILTER VALIDATION*

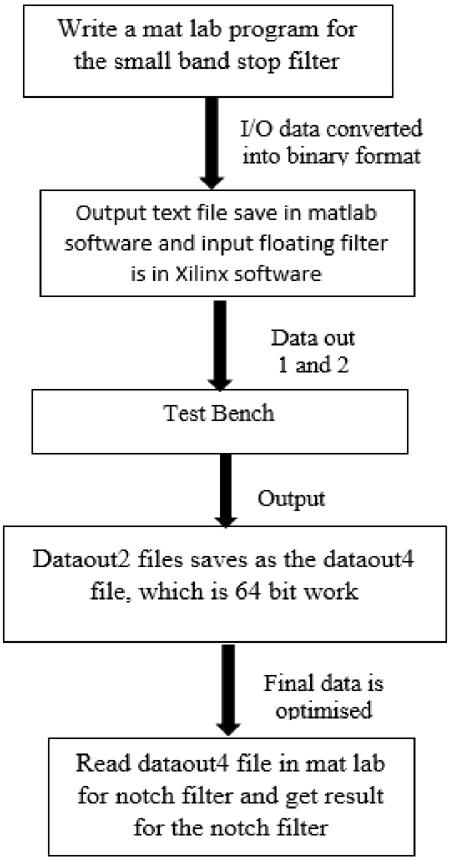

Write a mat lab program for the small band stop filter
I/O data converted into binary format
Output text file save in matlab software and input floating filter is in Xilinx software
Data out 1 and 2
Test Bench
Output
Dataout2 files saves as the dataout4 file, which is 64 bit work
Final data is optimised
Read dataout4 file in mat lab for notch filter and get result for the notch filter

**Figure-6:** Flow chart for validation of notch filters

Flow chart for validation of narrow band stop filter using MATLAB (for software implementation) and Xilinx ISE (for hardware implementation) is shown in figure 6. Various steps of flow chart are explained as follows:

- Write a MATLAB program for the small narrow band stop filter coefficient is calculate from all pass filter equations.
- Input data in real form and data output is in real format for the filter work for the real data only in MATLAB and then it converts into binary format by the MATLAB program.
- Output is taken from MATLAB and output file is generated and this file is save as text file which is the input for the floating filter in Xilinx software.
- Dataout2 files save as the dataout4 which is 64-bit format, and which work for input for the mat lab anti-notch filter then read the datoutput4 file in the MATLAB.
- Output is taken from MATLAB, which is same for the real and floating-point filter.
- Dataout2 files saves as the dataout4 file, which is 64-bit work for the input for the mat lab program for notch filter.

- Read dataout4 file in mat lab for notch filter and get result for the notch filter which is same for real and floating-point data for the filter.

### D. SIMULATION RESULTS

Prototyping of the notch filter in the MATLAB can be done by assuming an analog signal with following specifications:

- We take the sampling digital frequency is fs=1000 and t= 1/fs
- We take variable digital frequency is f1=100 and f2 =300 and f3 = 400
- Then combined these variable frequencies in the terms s=0.7*sin(2*pi*f1*t)+1*sin(2*pi*f2*t)+0.4*sin(2*pi*f3*t);
- Write real data input file in S in MATLAB

The frequency response of the IIR Notch filter is tuned to halt at $f_2$ = 300 Hz, as illustrated in figure 6.

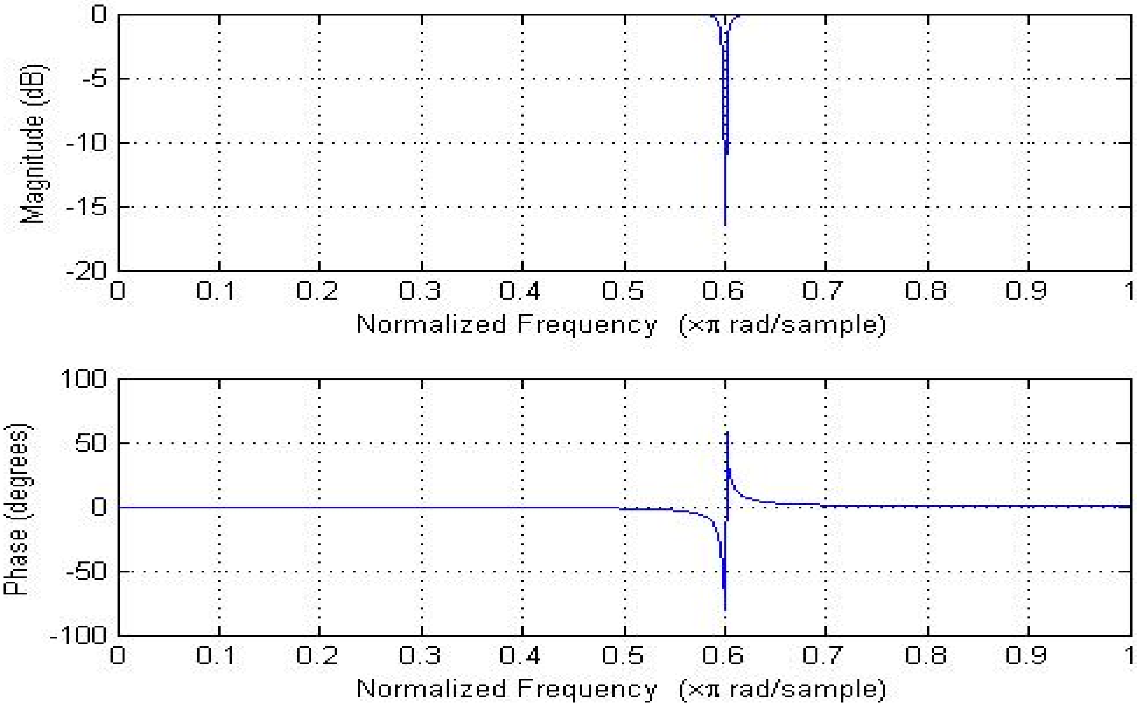
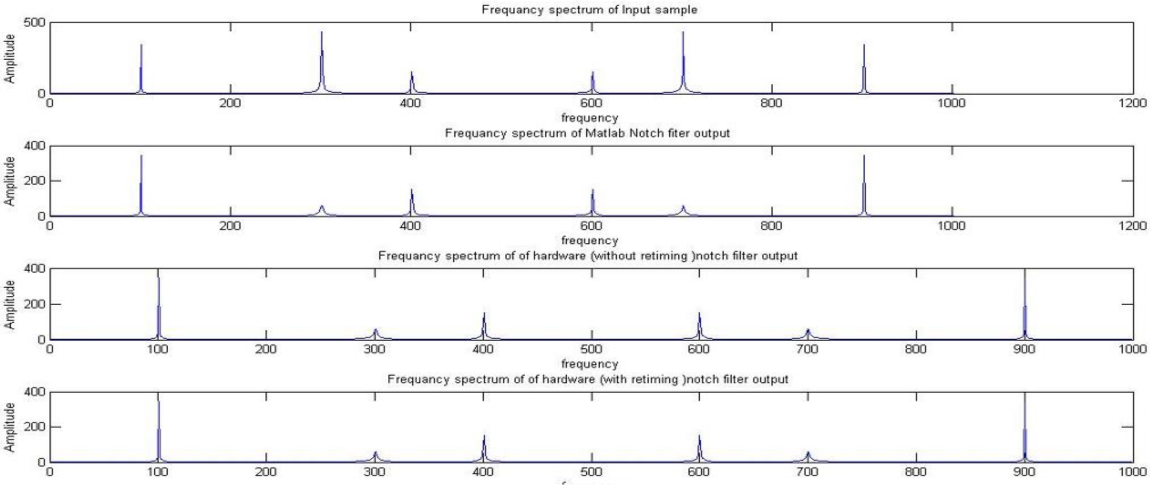


**Figure-7**: Frequency Response of IIR Notch Filter


Frequancy spectrum of Input sample

Frequancy spectrum of Matlab Notch fiter output

Frequancy spectrum of of hardware (without retiming )notch filter output

Frequancy spectrum of of hardware (with retiming )notch filter output

**Figure 8:** Output of IIR Notch Filter for MATLAB and Hardware Simulation

Additionally, MATLAB software simulates the suggested IIR notch filter for same input data. the output response of hardware and MATLAB simulation is shown in figure 7. these plots clearly indicate the same behaviour of stopping of the 300 Hz frequency component and passing the 100 and 400 Hz frequency component.

### *E. SYNTHESIS RESULTS*

Here HDL (VHDL) Language is used for RTL Coding. The Proposed IIR Notch Filter is Synthesized using Xilinx ISE 14.4 Version EDA tool and implemented on Nexys-4 DDR (Artix-7 FPGA Family) trainer kit.

**Table I:** Device Utilization Summary

| S. No. | Parameter | Used | Available |
|---|---|---|---|
| 1 | Number of slices | 132 | 35200 |
| 2 | Number of slices LUT | 2041 | 17600 |
| 3 | Number of unused flip-flops | 1991 | 2123 |
| 4 | Number with used LUT | 82 | 2123 |
| 5 | Percentages of LUT- FF pairs that are fully utilized | 50 | 2123 |

The device utilization summary is presented in table 1, which indicates the various components used of FPGA device. Timing report specifies the minimum clock period of 29.2 ns with its equivalent to 34.25 MHz at the maximum clock frequency.

## V. CONCLUSION

This paper proposes a notch IIR filter hardware implementation. Utilizing the Xilinx ISE Tool, the hardware architecture is created and then implemented on the Nexys- 4 DDR-FPGA (Artix - 7 family FPGA) Board. The proposed design has a critical path delay of 29.2 ns, which signifies the good performance of our design. Xilinx ISE Tool is used to implement the suggested design on an FPGA. This Notch filter is also validated using MATLAB for sampling frequency of 1000 samples/second with notch frequency of 300 Hz. The hardware implementation of narrow band stop filter signifies similar behavior compared to its equivalent MATLAB simulation. This proposed design has some limitations about its timing performance, which can be further increased by applying a retiming approach to its IIR direct form-II structure. The proposed design can also be extended by including the behavior of narrow band pass (NBP) filter in the same design along with the NBS filter.

## REFERENCES

[1] Jackson, Leland B. Digital Filters and Signal Processing: With MATLAB® Exercises. Springer Science & Business Media, 2013.

[2] Van de Plassche, Rudy J. Integrated analog-to-digital and digital-to-analog converters. Vol. 264. Springer Science & Business Media, 2012

[3] Ahn, Dal, et al. "A design of the low-pass filter using the novel microstrip defected ground structure." IEEE transactions on microwave theory and techniques 49.1 (2001): 86-93.

[4] Quatieri, Thomas F. Discrete-time speech signal processing: principles and practice. Pearson Education India, 2002.

[5] Tan, L., Jiang, J.: ‘Digital signal processing fundamentals and applications’ (Academic Press, Waltham, MA, USA, 2013, 2nd edn.

[6] Piskorowski, J.: ‘Digital Q-varying notch IIR filter with transient suppression’, IEEE Trans. Instrum. Meas., 2010, 59, (4), pp. 866–872

[7] Tan, L., Jiang, J., Wang, L.: ‘Pole radius varying IIR notch filter with transient suppression’, IEEE Trans. Instrum. Meas., 2012, 61, (6), pp. 1684–1691, Available at https://doi.org/10.1109/TIM.2012.2184013

[8] Rana, K.P.S., Markup, V., Gupta, A.: ‘A pole radius varying IIR notch filter with enhanced post transient performance’, Biomed. Signal Proc. Control, 2017, 33, pp. 379–391, Available at https://doi.org/10.1016/ j.bspc.2016.12.015

[9] Rana, K.P.S., Markup, V., Singhal, A., et al.: ‘Time varying pole radius IIR multi-notch filters with improved performance’, Arab. J. Sci. Eng., 2019, 44, (9), pp. 7101–7120, Available at https://doi.org/10.1007/s13369-019-03814-w

[10] Rathore, S. and Rawat, T.K., 2020, December. Design and Implementation of Time Varying pole Radius IIR Notch filter with improved performance. In 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN) (pp. 593-597). IEEE.

[11] Sharma, A., Kumar Rawat, T. and Agrawal, A., 2020. Design and FPGA implementation of lattice wave digital notch filter with minimal transient duration. IET Signal Processing, 14(7), pp.440-447.

[12] Vaidyanathan, P. P., and Byung-Jun Yoon. "Gene and exon prediction using all pass-based filters." Workshop on Genomic Sig. Proc. And Stat., Raleigh, NC. 2002.

[13] Vaidyanathan, P. P., and Byung-Jun Yoon. "Digital filters for gene prediction applications." Conference Record of the Thirty-Sixth Asilomar Conference on Signals, Systems and Computers, 2002. Vol. 1. IEEE, 2002

[14] Tseng, Chien-Cheng, and Soo-Chang Pei. "Stable IIR notch filter design with optimal pole placement." IEEE Transactions on Signal processing 49.11 (2001): 2673-2681.

[15] Mojiri, Mohsen, and Alireza R. Bakhshai. "An adaptive notch filter for frequency estimation of a periodic signal." IEEE Transactions on Automatic Control 49.2 (2004): 314-318.

[16] Vikas Pathak, Satyasai Jagannath Nanda, Amit Mahesh Joshi, Sitanshu Sekhar Sahu “High Speed Implementation of Notch/Anti-notch IIR Filter on FPGA”, 15th IEEE India Council International Conference (INDICON-2018), IEEE, Amrita University, Coimbtore, 2018, pp. 315-320.

[17] Vikas Pathak, Satyasai Jagannath Nanda, Amit Mahesh Joshi, Sitanshu Sekhar Sahu, “FPGA implementation of high-speed tunable IIR band pass notch filter for identification of hot spots in protein”, International Journal of Circuit Theory and Applications, 49.11 (2021): 3748-3765.

[18] Vikas Pathak, Satyasai Jagannath Nanda, Amit Mahesh Joshi, Sitanshu Sekhar Sahu, “Hardware Implementation of infinite impulse response Anti-notch Filter for Protein Coding Regions Identification in Eukaryotic Genes”, International Journal of Circuit Theory and Applications, 48.12 (2020): 2242-2256.