SSURJET

Sir Syed University Research Journal of Engineering & Technology
2022, Vol. 12, No. 1,
https://doi.org/10.33317/ssurj.414

Design and Performance Analysis of Improved FIR Filter using Ultra-Scale FPGA

Bhagwan Das[1], Javed Ali[1], Mahendar Kumar[2], Dileep Kumar[3], and Muhammad Zakir Shaikh[4]

[1]*Department of Electronic Engineering, Quaid-e-Awam University of Engineering, Science and Technology, Nawabshah, Sindh, Pakistan*
[2]*Department of Electrical Engineering, Mehran University of Engineering and Technology, Jamshoro, Sindh, Pakistan*
[3]*Department of Electronic Engineering, Dawood University of Engineering and Technology, Karachi, Pakistan.*
[4]*Department of Electronic Engineering, Mehran University of Engineering and Technology, Jamshoro, Sindh, Pakistan*

Correspondence Author: Bhagwan Das (engr.bhagwandas@hotmail.com)

Received December 29, 2021; Revised January 13, 2022; Accepted February 06, 2022

Abstract

It is discussed in many studies and demonstrated in many pieces of research that based on certain applications, analog design of filter has several issues including complex design, re-use limitations, and accuracy of generating the output at various frequencies. Therefore, instead of analog filter design, the digital design of the filter is preferred for both Finite and Infinite Impulse Response Filter. This paper demonstrates the design of the digital Finite Impulse Response (FIR) filter designed is demonstrated using Ultra-Scale Field Programming Gate Array (FPGA) having chip XCKU3P. The filter is designed using a coefficient multiplier via Canonic Signed Digit (CSD) Technique. The optimized design of the digital filter is conducted via real-time implementation is performed using Ultra-Scale FPGA. The filter is designed and evaluated with an ordinary filter at 10 MHz and 10 GHz frequencies. The performance analysis of the system is illustrated using the response rate at the bitstream of 16-bit. In the results, it is demonstrated that for 10 MHz frequency design FIR filter in FPGA the 30% faster response filter is achieved at for 10 GHz, the 15% faster response is achieved at the I/O standard of Low Voltage Complementary Metal Oxide Semiconductor (LVCOMS). The optimization of 30% in terms of the response time of the filter is attained using the proposed work. The proposed improved FIR filter design using Ultra-Scale FPGA helps in increasing design performance to increase the speed of overall response of FIR filter that is lacking in ordinary Filters.

Index Terms: *Canonic Signed Digit Technique, Digital Filter Design, Filter Response, Response Rate, Ultra-Scale FPGA.*

I. INTRODUCTION

In signal processing, filter design is an important application for various design applications during that many design procedures are referred. The filter design is conducted using analog and digital design. The filter realization is conducted for different design applications [1]. The filter design is evaluated using the different parameter characterizations such as response, stability, and other features [2]. The response of the filter defines the rate at which output is attained after design. In many applications, the response of the filter is the most deciding factor [3]. The filters are categorized using low, high, and other design aspects [4]. It is defined that in many aspects the filter design in real-time is more important than the simulation design [5]. In this domain, many approaches are utilized such as filter design using ICs, and DPS boards are most common [6]. Among these, the filter design using Field Programming Gate Array (FPGA) is widely used recently [7]. As the FPGA design of filter provides the various advancements such as optimization, performance analysis, and many more.

In this work, the optimized digital Finite Impulse Response (FIR) filter design is demonstrated using the proposed technique and the filter is implemented in Ultra-Scale FPGA. The designed filter is fast in response to both low and high frequencies. In the next, the research methodology, results discussion, and the conclusion is illustrated. The reason for the low Pass is based on that the application of this work will be at the receiver side used in digital communication.

II. LITERATURE REVIEW

There are numerous studies are published related to filter optimization and some of them related to this work are discussed as:

In 2016 the author Zhao in the proposed design of FIR filter using Mean Square Error (MSE). The results discussed in this research no longer be at cross purposes because the filter is shown to be a special case of the algorithm, which does not require noise statistics.

In 2015 the authors Liu and associates discussed the modern design of linear phase FIR using logic gates. The optimization is missing in this research.

The author Pak J M in 2016 developed a self-recovering algorithm. The work was based on tracking problems. The author Bhaskar worked on Distributed Arithmetic (DA) in FIR filter is replaced with a multiplier to remove high-frequency noise from ECG signal. The author Anmin compared the linear phase-frequency characteristic of Infinite Impulse Response (IIR) digital filter, Finite Impulse

Response (FIR) digital filter not only ensures the accurate strict linear phase characteristic, simple structure, and stability in the design and application of the digital filter. The author Sun Chao in 2020 designed by the analog circuit the implementation was conducted on FPGA. This paper explains the FIR filter principle and filter coefficient calculation and simulates the linear filter with the sampling rate of 1 MHz, the passband is 120 kHz, and the order is 15^{th} order. Finally, the experimental verification is carried out through the FPGA board, and the corresponding design. The system was designed using low parameters without optimization.

In this work, the optimization of the filter in terms of response time and power is conducted. In the first, the filter is designed in MATLAB and its optimization is conducted using the Canonic Signed Digit (CSD) technique after that design in FPGA is demonstrated and tested on various high frequencies to validate the performance of the designed optimized filter using FPGA.

III. RESEARCH METHODOLOGY

In this work, the optimized digital filter design is carried out via real-time implementation is carried out Ultra-Scale FPGA. The design methodology is supported using the following design steps. The design methodology is depicted in Figure 1. In the first design step, an optimized digital FIR filter is designed using MATLAB. In the second design step, quantization of the FIR filter is performed. In the third design step, the 'Configuration' and 'Generation' of optimized 'Verilog Code' are performed.

In the fourth design step, exploration of 'Explore the Optimized Generated Verilog Code' and in the last design step, the development of 'Generated Verilog Code' is executed using Ultra-Scale FPGA. In the next, each design step is discussed in detail.

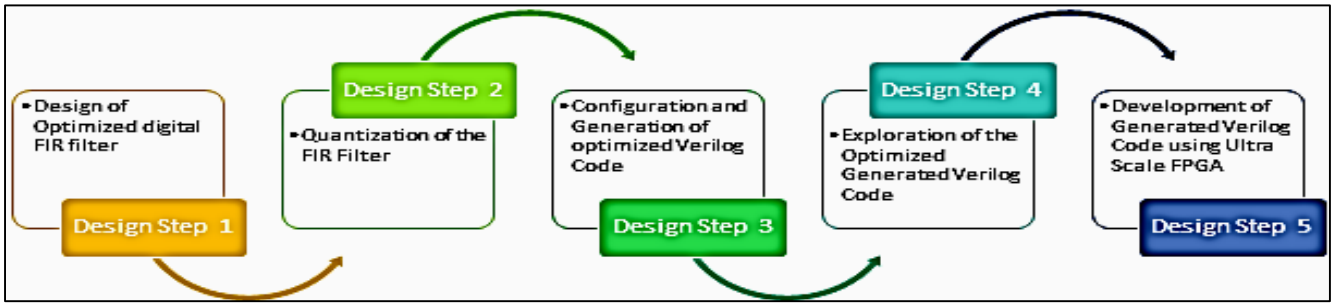

Figure 1: Design Methodology

A. Design Step 1: Design of Optimized FIR Filter

The filter is designed using the 'Filter Designer' tool placed in the 'Filter Design and Analysis Tool' command in MATLAB. The dialog box appears. The filter is designed using the specification as shown in Table 1.

Table 1: Optimized Filter Specification [8]

Filter Specifications	Values	
Response Type	Lowpass	
Method of Design	FIR Equiripple	
The Oder of Filter		
Density Factor	20	
Frequency Specification	Sampling Frequency	48000 Hz
	Pass Frequency	9600 Hz
	Stop Frequency	12000 Hz
Magnitude Specification	Amplitude Pass	1 dB
	Amplitude Stop	80 dB

B. Design Step 2: Quantization of FIR Filter

It is defined that after the optimized filter design in step 1, it is mandatory to quantize the filter, to generate the Hardware Description Language (HDL) code for the filter. Because the FPGA is working on digital samples of input. To quantize the filter, have to access the 'Filter Design tool' and click on the quantization button. In the quantization, the filter arithmetic design is selected. The filter arithmetic offers the quantization using Fixed-point with Filter precision [9].

The quantization of the optimized filter is attained using the following design specification in order to achieve high performance and accuracy. The design specifications are discussed using Table 2.

Table 2: Specification of Quantized and Optimized Filter

Specification for Quantization	Values
Coefficients for Numerator Word Length	16
Input and Output Word Length	16
Filter Internals Rounding Mode	Floor
Overflow Mode	Saturate
Accum. Word Length	40

In the next, the quantized filter is configured to generate the optimized 'Verilog Code' that will be executed on Ultra-Scale FPGA. To generate the HDL code, select the 'Target' and select the 'Generate HDL' in the toolbox.

After that select, the Verilog option [10]. The HDL configuration is attained using an optimized filter design that will be optimized for performance or space requirements. The reason for selecting this is the coder makes tradeoffs concerning data types and might ignore your quantization settings to achieve optimizations.

C. Design Step 3: Configuration and Generation of Optimized Verilog Code Using CSD

The optimized filter design using HDL code is designed and developed using Canonical Signed Digit (CSD) technique. CSD is a sophisticated technique to design the digital using digital multipliers. The filter is designed using symmetry in the filter coefficients that is attained using 16 multiplications to design filter output. CSD is faster than the exiting multiplier. Here it is discussed how CSD is working. The CSD is demonstrated using the CSD design as shown in Figure 2 [11].

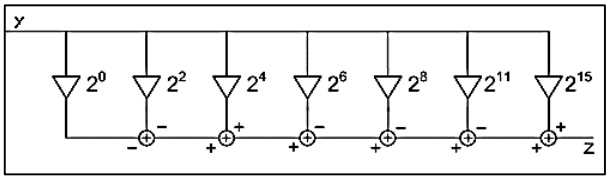

Figure 2: CSD Design Methodology

The proposed CSD technique for the optimization filter is utilized. The technique is used to implement multiplication by a fixed coefficient, here in this work 0.34 and after that as scaling factor is applied to develop the rounded coefficient.

It is important to note that the scaling factor is selected via the magnitude of the largest coefficient and the 16 bits for defining the coefficients. The scaling factor is selected as 10^{35} means to implement the round coefficient 34359738368. That will have produced the 10000000000000000000000000000000000. The CSD representation of this coefficient is 10001001010101011000100101010101. The design is demonstrated using Figure 3 which defines the extra circuitry to convert the binary number into CSD [12].

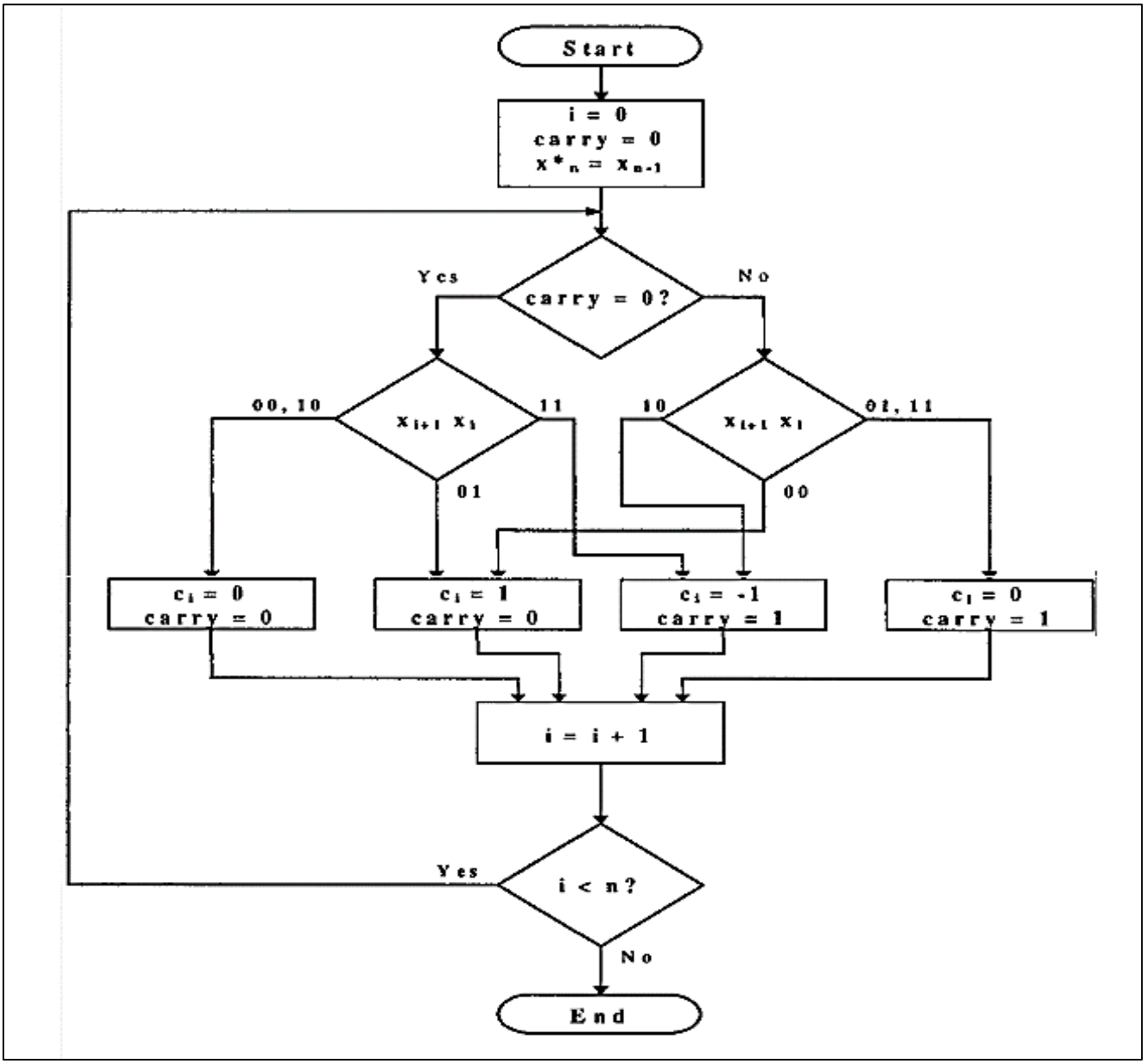

Figure 3: CSD based Algorithm for Optimized Filter Design

The MATLAB design of Configuration and Generation of optimized Verilog Code using CSD is designed and demonstrated. CSD technique minimizes the number of addition operations required for constant multiplication by representing binary numbers with a minimum count of nonzero digits.

To execute the design in MATLAB, select the 'Add' pipeline registers for optimized final summation. In the next, 'Global' settings is selected after that generate 'Test Bench'. In the 'Test Bench' pane, observe that the 'Error Margin' (bits) option is enabled. In the next, the code is explored for the testing of Optimized Generated Verilog Code.

D. Design Step 4: Exploration of the Optimized Generated Verilog Code

The code is explored using 'optfir.v' in an ASCII or HDL simulator editor. The design defines the 'Ports for Generating HDL' with the values you specified for the 'Input port' and 'Output ports.
The block name Delay-Pipeline-process includes the default block postfix '_process'. The block name sumdelay_pipeline_process1. In the end, data-out is produced.

E. Design Step 5: Development of Generated Verilog Code using Ultra-Scale FPGA

The design and development of optimized Verilog Code using CSD for FIR Filter are demonstrated using Ultra-Scale FPGA. For this load, the Verilog Code in Xilinx ISE as a folder that contains your generated Verilog files. Create a design library with the 'Vlib Command'. Compile the generated filter and test bench Verilog files. Load the test bench for simulation in the design specification. Figure 4 shows the Ultra-Scale FPGA design of the optimized FIR filter.

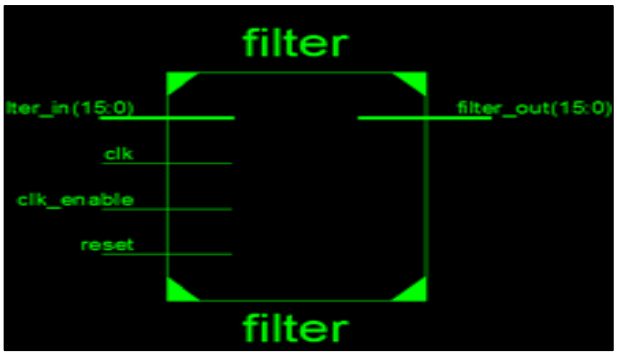

Figure 4: Ultra-Scale FPGA Optimized Filter Design

The Ultra-Scale FPGA design of the optimized FIR filter is demonstrated in test bench simulation verification. If error messages appear, interpret them as they pertain to your filter design and the HDL code generation options you selected. Determine whether the results are expected based on the customizations you specified when generating the filter Verilog code. In the next, the designed Ultra-Scale FPGA design of optimized FIR filter is tested for performance analysis [13-15].

IV. RESULTS AND DISCUSSION

The designed system offers the optimized filter design using the proposed CSD technique and its design using Ultra-Scale FPGA. The results are discussed in the following manner that in the first the response of ordinary and CSD-based optimized filter is shown and after that response of with and without quantization is shown. After that Register Transfer Level (RTL) design of the proposed CSD technique and its design using Ultra-Scale FPGA is shown and at the end response in terms of timing analysis and other parameters are demonstrated. Figure 5 shows the response of the ordinary response of the FIR filter without using the proposed technique Canonical Signed Digit (CSD) i.e., the response is without optimization of the filter. It shows that the filter is Low pass and configured for minimum order with 48000 Hz. It can be seen that rippers are observed in the response.

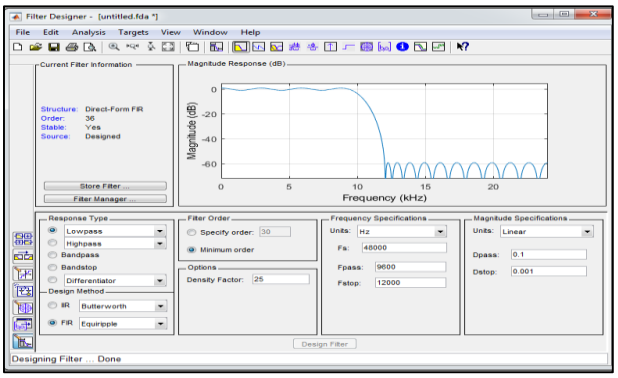

Figure 5: Design Response of Ordinary FIR Filter

The response of optimized filter design using the proposed CSD is shown in Figure 6. The optimized filter designed using the proposed technique CSD has attained the smooth response and using our proposed technique CSD the optimized order of the filter is calculated that is 35.

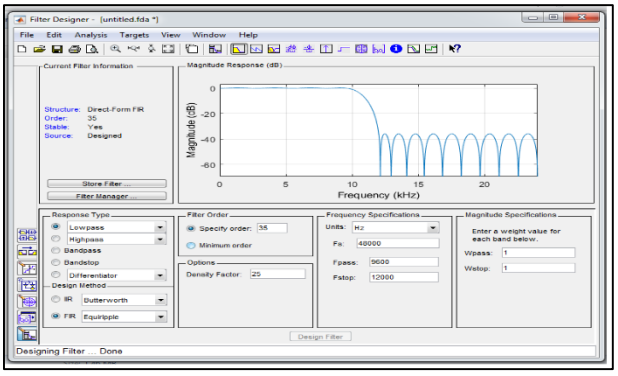

Figure 6: Design Response of Optimized FIR Filter

The filter is designed using MATLAB via the 'Filter Design and Analysis Tool'. The reason for selecting the 'Equiripple' method for filter design is that it is a good and reliable method for approximation to the desired frequency response. The significant benefit is the Fast Fourier Transform (FFT)-based design method is its implementation because of ease and versatility. This is why the optimized filter is mentioned to attain the best response for the filter design. The response of the non-quantized filter is shown in Figure 7. The Non-quantized filter is missing from arithmetic design aspects as the non-quantized filter doesn't have Fixed-point with Filter precision.

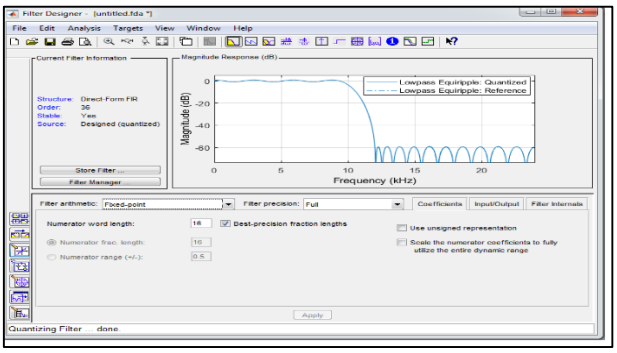

Figure 7: Design Response of Non-quantized Optimized Filter

The quantization of an optimized filter is important to generate the HDL code for the filter as FPGA is working on digital samples of input. The quantization is achieved using the 'Filter Design' tool. The quantization offers the quantization using Fixed-point with Filter precision. The quantization of the optimized filter is attained using the following design specification to achieve high performance and accuracy. The response of the designed quantized optimized filter is shown in Figure 8.

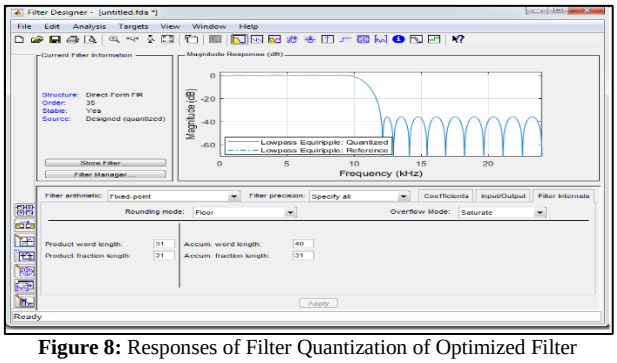

Figure 8: Responses of Filter Quantization of Optimized Filter

The designed optimized and quantized filter is designed and developed in FPGA using Verilog code. In the next, the quantized filter is configured to generate the optimized Verilog code that will be executed on Ultra-Scale FPGA. To generate the HDL code, select the 'Target' and select the 'Generate HDL' in the toolbox. After that select, the 'Verilog' option. The HDL configuration is attained using an optimized filter design that will be optimized for performance or space requirements. The reason for selecting this is the coder makes tradeoffs concerning data types and might ignore your quantization settings to achieve optimizations. The RTL schematic for the optimized filter has been demonstrated as shown in Figure 9.

The RTL of the designed optimized filter having RTL using VHSIC Hardware Description Language (VHDL) that is implemented on Ultra-Scale FPGA is demonstrated in Figure 9. It is important to note that designed the optimized filter having RTL using VHDL that is implemented on Ultra-Scale FPGA is designed using various design parameters such as input data, output data, clock as input, an enable signal for the clock, and the reset option.

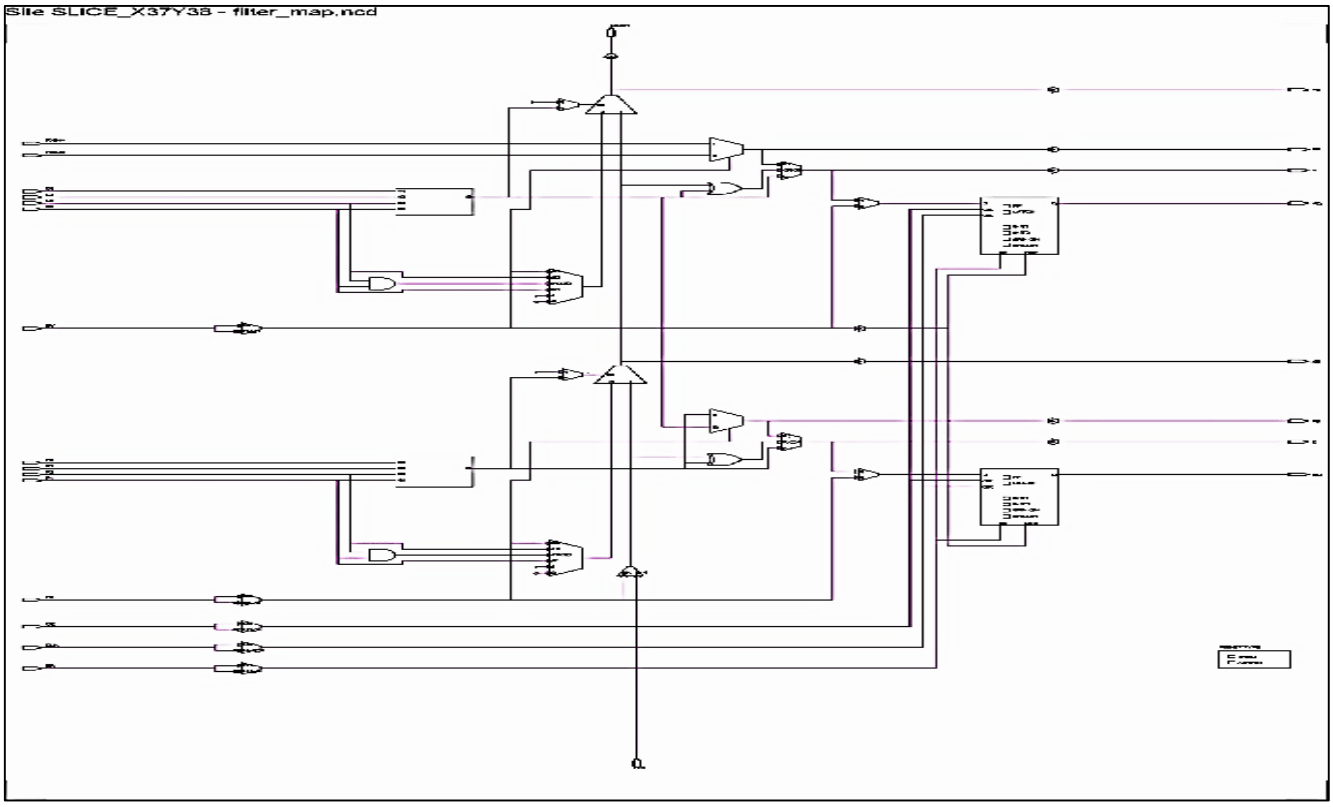

Figure 9: RTL Design of Optimized Filter

The filter is designed for a 16-bit configuration as an input bitstream and the same as an output bitstream. One of the unique features of the destined filter having RTL using VHDL is that failure test is also designed for the system, if any failure in the design is observed the signal will be high. The system is also designed for the test bench delay and various other parameters. In the next, the response of the without optimized design and optimized using the proposed technique is observed. In Figure 10 response of the filter is demonstrated. The response is based on ordinary filter design in Ultra-Scale FPGA. It means the response is shown without the proposed technique. The response illustrated that 'Clk Pulse' is fixed and clock sign is enabled, and the reset option is presented and will be active high whenever is required. Filter input of 16-bit is inserted in the design and similarly, the output is also 16-bit. It is important to note that the input response is delayed from the output response, and it is calculated that a delay of 20 microseconds is observed. The 'tb_enb' signal defines the artificial delay tab to include based on insertion. The 'srcDone' defines the sample rate conversion that is not set now and can be programmed as per need. The 'snkDone'

represents the synchronous activation and for this, it is already programmed and can be modified at any time. The 'TestFailure' option is low activated as there no error or fault is found. The 'tbenb_dly' is activated as it tends to be high because of the proper operation of the delay signal. Similarly, the 'rdenb' is active high due use of standard logic IO Standard, and in this case by default IO Standard is used that is LVCMOS 1.5V. The filter_in_data_log_rdenb defies the active high signal due to activating the option of reading enable if data bits are changed need to update the filter accordingly. Filter_in_data_log_addr [11:0], this command writes the address of memory at which function stores the data. Filter_in_data_log_done, this commanded keeps all the entries of the memories that are updated in the filer design. Filter_out_testfailure, this active low signal as there is no test failure response from the filter design. Filter_out_errcnt is again a zero data signal as there is no error found during the filter testing. The delayline_out is an active high signal as it is consecutively checking the delay incurred in transmitted and received pulses. The expected_ce_out signal is associated with delayline as during the delay line check the signal remains active high to read the clock enable for the excepted delay line. After that the pulse response of filter out is demonstrated the filter_out_rdenb is active high signal as the signal is high the outcome read enable will initiate the filter to generate the response. The filter_out_addr is an address at which the memory location is reserved to store the outcome of the filter. Filter_out_done is a low signal as there is no error at output. After that using filter_out_ref [15:0] 16-bit data reaches out at the output pulse. After that clock signal is descripted that tells the clock signal of 5000 ps is kept high and the 5000 ps signal is kept low and Clk_period is 10000 ps the max_error_count in terms of clock pulse will be the same as input data as there is no error is found.

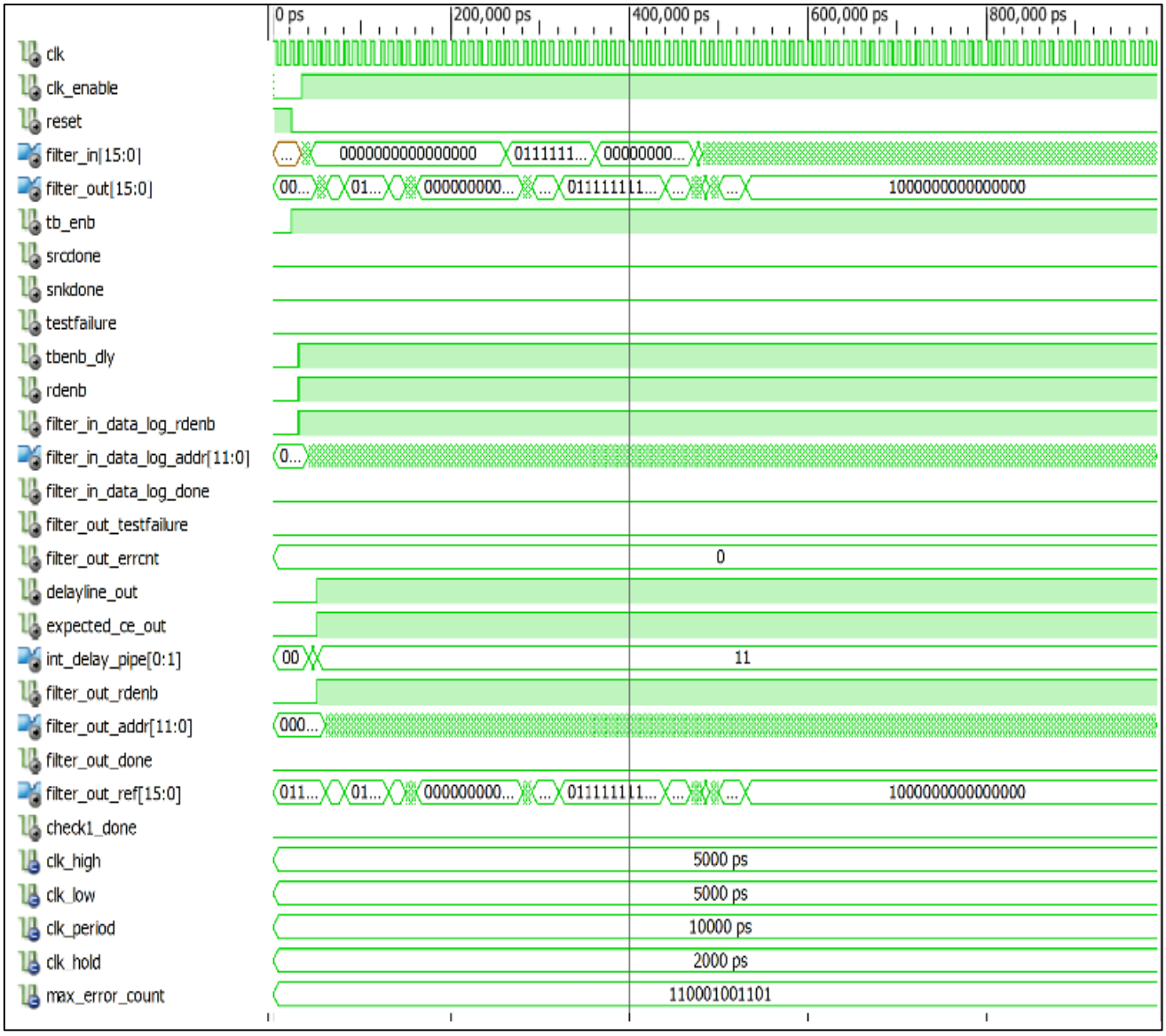

Figure 10: Response of Filter without Implementing Proposed Technique

Figure 11 defines the response of the proposed filter design using the designed technique in Ultra-Scale FPGA. It means the response is shown with the proposed technique. The response illustrated that 'Clk pulse' is fixed and clock

sign is enabled, and the reset option is presented and will be active high whenever is required. Filter input of 16-bit is inserted in the design and similarly, the output is also 16-bit. It is important to note that the input response is delayed from the output response, and it is calculated that a delay of 5 microseconds is observed.

The tb_enb signal defines the artificial delay tab to include based on insertion. The srcDone defines the sample rate conversion that is not set now and can be programmed as per need. The snkDone represents the synchronous activation and for this, it is already programmed and can be modified at any time. The testfailure option is low activated as there no error or fault is found.

The tbenb_dly is activated as it tends to be high because of the proper operation of the delay signal.

Similarly, the rdenb is active high due use of standard logic IO Standard, and in this case by default IO Standard is used that is LVCMOS 1.5V. The filter_in_data_log_rdenb defies the active high signal due to activating the option of reading enable if data bits are changed need to update the filter accordingly. Filter_in_data_log_addr [11:0], this command writes the address of memory at which function stores the data. Filter_in_data_log_done, this commanded keeps all the entries of the memories that are updated in the filer design. Filter_out_testfailure, this active low signal as there is no test failure response from the filter design. Filter_out_errcnt is again a zero data signal as there is no error found during the filter testing. The delayline_out is an active high signal as it is consecutively checking the delay incurred in transmitted and received pulses. The expected_ce_out signal is associated with delayline as during the delay line check the signal remains active high to read the clock enable for the excepted delay line.

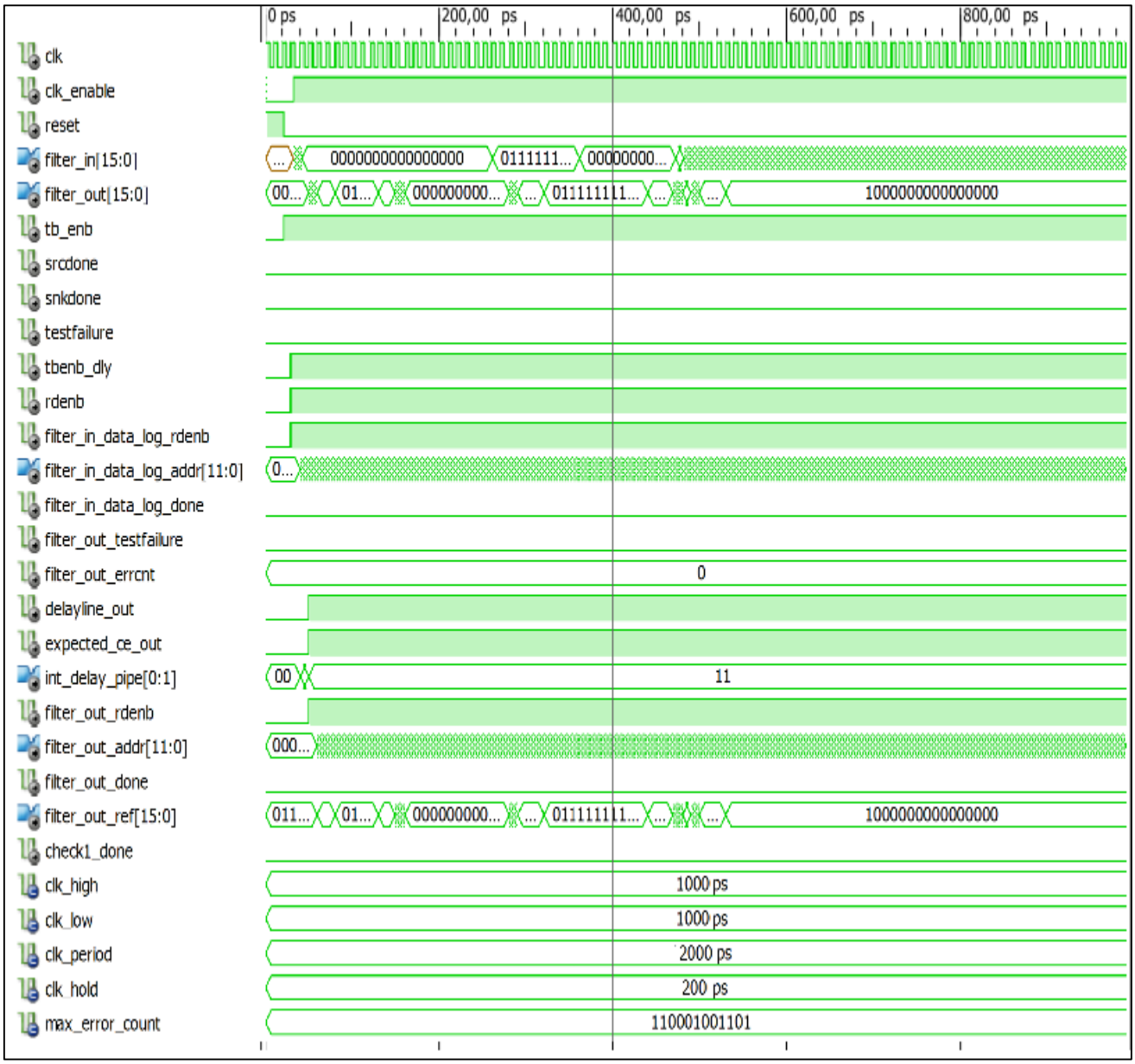

Figure 11: Response of Filter without Implementing Proposed Technique

After that the pulse response of filter out is demonstrated the filter_out_rdenb is active high signal as the signal is high the outcome read enable will initiate the filter to generate the response. The filter_out_addr is an address at which the memory location is reserved to store the outcome of the filter. Filter_out_done is a low signal as there is no error at output. After that using filter_out_ref [15:0] 16-bit data reaches out at the output pulse. After that clock signal is descripted that tells the clock signal of 1000 ps is kept high and the 1000 ps signal is kept low and Clk_period is 1000 ps the max_error_count in terms of clock pulse will be the same as input data as there is no error is found. It can be compared with Figure 10 and Figure 11 that response of filter design of proposed CSD technique in the Ultra-Scale FPGA is faster than the ordinary filter designed in FPGA. It is demonstrated that the filter is 30% faster in response in terms of less delay is demonstrated and it is defined that no error is found.

Figure 12 defines the response of the proposed filter design using the designed technique in Ultra-Scale FPGA. It means the response is shown with the proposed technique. The response illustrated that the Clk pulse is fixed, and clock sign is enabled, and the reset option is presented and will be active high whenever is required.

Filter input of 16-bit is inserted in the design and similarly, the output is also 16-bit. It is important to note that the input response is delayed from the output response, and it is calculated that a delay of 10 microseconds is observed. The tb_enb signal defines the artificial delay tab to include based on insertion.

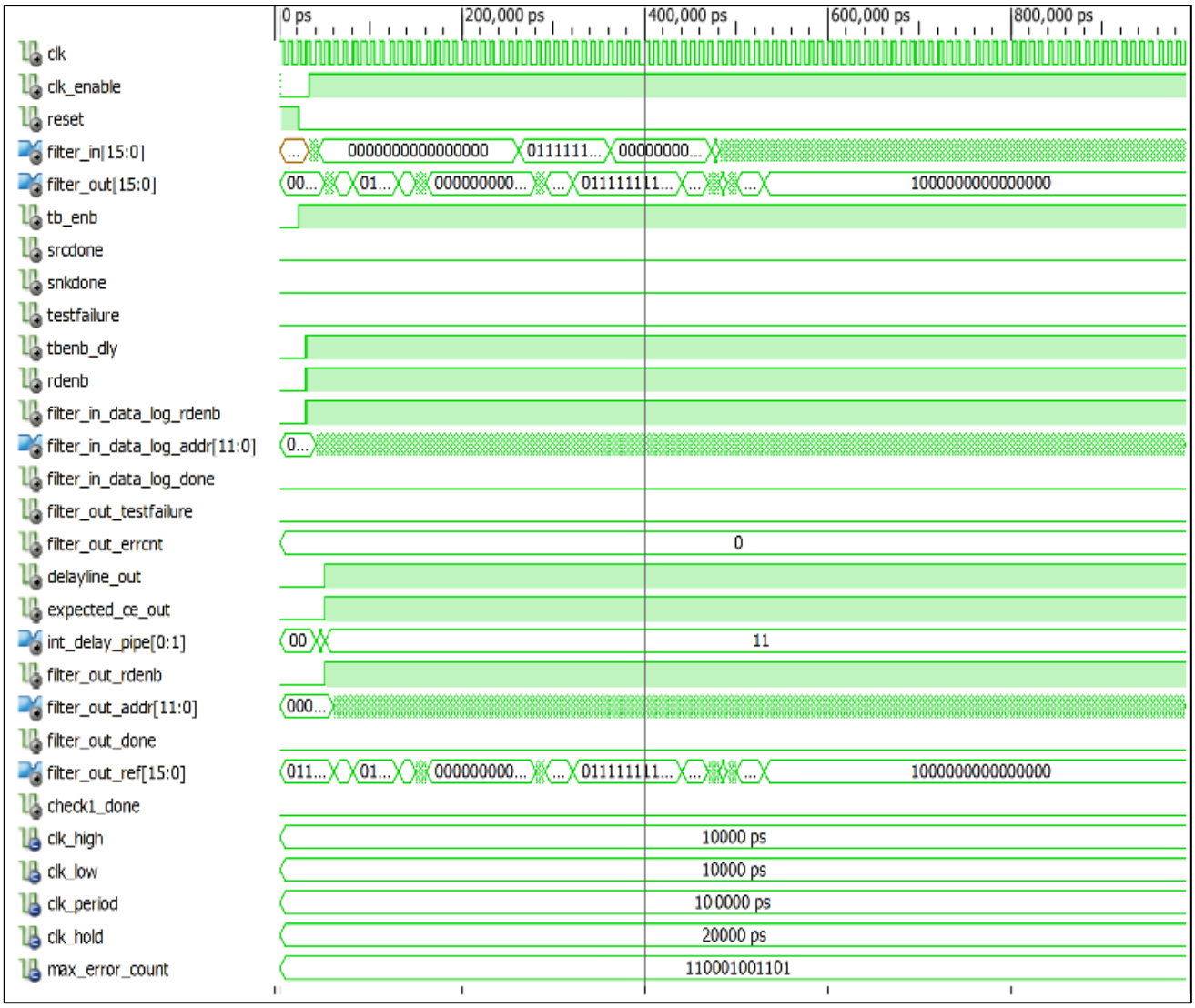

Figure 12: Response of Filter without Implementing Proposed Technique

The srcDone defines the sample rate conversion that is not set now and can be programmed as per need. The snkDone represents the synchronous activation and for this, it is already programmed and can be modified at any time. The testfailure option is low-activated as there no error or fault is found. The tbenb_dly is activated as it tends to be high because of the proper operation of the delay signal. Similarly, the rdenb is active high due use of standard logic IO Standard, and in this case by default IO Standard is used that is LVCMOS 1.5V. The filter_in_data_log_rdenb defies the active high signal due to activating the option of reading enable if data bits are changed need to update the filter accordingly. Filter_in_data_log_addr [11:0], this command writes the address of memory at which function

stores the data. Filter_in_data_log_done, this commanded keeps all the entries of the memories that are updated in the filer design. Filter_out_testfailure, this active low signal as there is no test failure response from the filter design. Filter_out_errcnt is again a zero data signal as there is no error found during the filter testing. The delayline_out is an active high signal as it is consecutively checking the delay incurred in transmitted and received pulses. The expected_ce_out signal is associated with delayline as during the delay line check the signal remains active high to read the clock enable for the excepted delayline. After that the pulse response of filter out is demonstrated the filter_out_rdenb is active high signal as the signal is high the outcome read enable will initiate the filter to generate the response. The filter_out_addr is an address at which the memory location is reserved to store the outcome of the filter. Filter_out_done is a low signal as there is no error at output. After that using filter_out_ref [15:0] 16-bit data reaches out at the output pulse. After that clock signal is descripted that tells the clock signal of 10000 ps is kept high and the 1000 ps signal is kept low and Clk_period is 10000 ps the max_error_count in terms of clock pulse will be the same as input data as there is no error is found. It can be compared with Figure 11 and Figure 12 that response of filter design of proposed CSD technique in the Ultra-Scale FPGA is faster than the ordinary filter designed in FPGA. It is demonstrated that the filter is 20% faster in response in terms of less delay is demonstrated and it is defined that no error is found.

V. CONCLUSION

In this work, the FIR filter is designed using the CSD technique, and the filter is implemented in the FPGA. It is demonstrating that the filter is 30% faster in response in comparison to order frequency at 10 MHz operating frequency. There is more frequency is increased at 10 GHz and it is demonstrated that 10% faster than the ordinary design of the filter. The filter is designed in Ultra-Scale FPGA using LVCMOS IO Standard and in the future, a more optimized design can be developed to have a more efficient and fast response filter by varying the IO Standards. The proposed optimized filter will be used in a high-speed digital communication system to provide reliable communication.

Acknowledgement

The authors would like to thank; **Quaid-e-Awam** University of Engineering, Science and Technology, Nawabshah, Sindh, Pakistan, **Mehran** University of Engineering and Technology, Jamshoro, Sindh, Pakistan, and **Dawood** University of Engineering and Technology, Karachi, Pakistan for all the support provided to accomplish this research work.

Authors Contributions

Bhagwan Das contribution to this study was the concept, administration, and correspondence. Javed Ali, Dileep Kumar, and Muhammad Zakir Shaikh performed data collection, supervision, and methodology. They also performed the data compilation and validation. Mahendar Kumar contributed to project administration and paper writing.

Conflict of Interest

There is no conflict of interest between all the authors.

Data Availability Statement

The testing data is available in this paper.

Funding

This research received no external funding.

References

[1] Koseoglu, M., Deniz, F. N., Alagoz, B. B., & Alisoy, H. (2021). An effective analog circuit design of approximate fractional-order derivative models of M-SBL fitting method. *Engineering Science and Technology, an International Journal*.

[2] Subramaniam, B., Siddik, Z. H., & Nagoor, N. H. (2020). Optimization of nanostructured lipid carriers: Understanding the types, designs, and parameters in the process of formulations. *Journal of Nanoparticle Research, 22*, 1-29.

[3] Böhler, L., Ritzberger, D., Hametner, C., & Jakubek, S. (2021). Constrained extended Kalman filter design and application for online state estimation of high-order polymer electrolyte membrane fuel cell systems. *International Journal of Hydrogen Energy, 46*(35), 18604-18614.

[4] Marinescu, R. (2004, September). Detection strategies: Metrics-based rules for detecting design flaws. In *20th IEEE International Conference on Software Maintenance, 2004*. Proceedings. (pp. 350-359). IEEE.

[5] Roy, T., & Bhattacharjee, P. (2020). A LabVIEW-based real-time modeling approach for detection of abnormalities in cancer cells. *Gene Reports, 20*, 100788.

[6] Kirkpatrick, D. A., & Diduck, Q. (2021). U.S. Patent Application No. 17/369,676.

[7] Sudharsan, R. R., & Deny, J. (2020). Field programmable gate array (FPGA)-based fast and low-pass finite impulse response (FIR) filter. In *Intelligent Computing and innovation on data science* (pp. 199-206). Springer, Singapore.

[8] Gao, Y., Zhang, F., Lv, X., Guo, C., Shang, X., Li, L., ... & Lancaster, M. J. (2020). Substrate integrated waveguide filter–amplifier design using active coupling matrix technique. *IEEE Transactions on Microwave Theory and Techniques, 68*(5), 1706-1716.

[9] Ahmed, A., Hussain, G., & Raza, A. (2021). Ultra-Wide Band Horseshoe Antenna for Cognitive Radio Applications. *Journal of Applied Engineering & Technology (JAET), 5*(1), 51-60.

[10] Borges, V., Nepomuceno, E. G., Tutueva, A. V., Karimov, A. I., Duque, C., & Karimov, T. I. (2020, March). Analysis of IIR Filters by Interval Response. In *2020 Moscow Workshop on Electronic and Networking Technologies (MWENT)* (pp. 1-5). IEEE.

[11] Cai, L., Qian, Y., He, Y., & Feng, W. (2021, May). Design of Approximate Multiplierless DCT with CSD Encoding for Image Processing. In *2021 IEEE International Symposium on Circuits and Systems (ISCAS)* (pp. 1-4). IEEE.

[12] Kumar, G. K., & Narayanam, B. (2020). Low Power Implementation of FIR Filter for De-noising the EOG signal. *International Journal of Computing and Digital Systems, 9*(5), 965-970.

[13] Ahmed, S., & Naseem, M. (2014). 2 Efficient AES-XTS Pipelined Implementation on FPGA. *Sir Syed University Research Journal of Engineering & Technology, 4*(1), 6-6.

[14] Siddiqui, N. A., Siddique, A. A., Farooq, F., & Qadri, M. T. (2018). 1 Gesture-Based Communication System for Vocally Impaired. *Sir Syed University Research Journal of Engineering & Technology, 8*(1), 4-4.

[15] Das, B., Abdullah, M. F. L., & Pandey, B. (2019). Low Power Design of 40 Gigabit Ethernet Media Access Controller Using Hyper Transport Protocol IO Standard. *Sir Syed University Research Journal of Engineering & Technology, 9*(1).