Journal of Engineering Science and Technology
Vol. 20, No. 1 (2025) 345 - 361

FPGA IMPLEMENTATION OF LOWPASS FIR FILTER USING SINGLE MULTIPLY-ACCUMULATE UNIT WITH DUAL-PORT RAM

AMER KAIS ALJUMAILI[1],
RAAED F. HASSAN[2], EKHLAS K. HAMZA[1], AMJAD J. HUMAIDI[1,*]

[1]Control and Systems Engineering Department, University of Technology, Baghdad, Iraq
[2]Electrical Engineering Technical College, Middle Technical University, Baghdad, Iraq
*Corresponding Author: amjad.j.humaidi@uotechnology.edu.iq

Abstract

Finite Impulse Response (FIR) filters are crucial in advanced systems such as Digital Signal Processing (DSP) and image processing, thanks to their linear phase properties and stability. However, attaining superior performance demands the implementation of higher-order FIR filters, leading to more evaluation processes, greater hardware needs, and elevated energy consumption. To address these issues, a single Multiply Accumulate (MAC) unit with addressable dual-port RAM has been suggested in this paper to create a symmetric FIR filter. The proposed structure has been set up to have the benefit of being able to create any kind and degree of FIR filter. The filter coefficients are extracted in this paper using the unconstrained least-Pth norm optimization approach using MATLAB. Field Programmable Gate Array (FPGA) technology was applied to accomplish the suggested FIR filter arrangement based on Xilinx System Generator (XSG). The frequency and time responses of the designed FIR filter were validated by contrasting its operation with those of the reference FIR filter using simulation. Real-time Hardware-In-Loop (HIL) co-simulation was executed while the proposed filter was implemented on a Xilinx Spartan6 FPGA device (XC6SLX45T), which is part of the SP605 Evaluation kit. Based on the simulation results, the frequency response of the filter created using the given parameters closely aligns with that of the reference filter. Furthermore, real-time testing of the developed filter has been carried out, and the experimental results indicate that the desired frequency response has been achieved.

Keywords: FPGA, Lowpass FIR, Single multiply-accumulate, Dual-port RAM.

1. Introduction

Digital communication has undoubtedly been a significant achievement in the modern era, enabling the exchange of information faster and more effectively than ever before [1-4]. One of the challenges of digital communication is the noise in the received data [5]. To overcome the problem of noise in digital communication, effective filters need to be designed to remove or minimize the noise [6]. Filters are electronic circuits or algorithms that selectively allow certain frequencies to pass through while blocking others [7, 8].

Digital filters are classified into two types according to their impulse response: the first type is the Finite Impulse Response (FIR) filters, and the second type is the Infinite Impulse Response (IIR) filters FIR filters provide a weighted sum of the input and preceding input. The coefficients of the filter are determined according to its design. FIR filters have a smooth phase response, making them useful in applications requiring little phase distortion [9, 10], such as photography, audio, communications, control and biological signal processing. Their adaptability and ease of use make them essential tools for signal conditioning and filtering [11].

The filter order is directly proportional to the cost and computational latency of the filter. Filter order affects the frequency resolution and sharpness of the filter. Higher resolution filters provide better resolution and better accuracy, on the other hand the cost is the increased of complexity [12-18]. The purpose of reducing the coupling of FIR filters is to make them efficient and effective for real-time applications. By reducing computational complexity and still being accurate, filters can be implemented using fewer resources (such as hardware or computer systems) [19, 20]. This led researchers to focus on developing high-efficiency, low-energy filters. Therefore, this topic has received many important contributions from international researchers [19].

The complexity of the coefficient multiplier determines the complexity of the FIR filter, and the most popular solution that has emerged in the past is multiple cancellation (CSE) [21-25]. These algorithms are based on obtaining the FIR filter structure by dealing with fixed parameters, and they are time-consuming even when working on a very efficient computing platform. However, this approach is not suitable for many applications in which the filter coefficients are dynamically updated. Recently, several techniques have been suggested to reduce the number of multiplication and addition units while maintaining the efficiency of the FIR filter [20].

The backbone of the digital FIR filter structure is the multiply accumulator (MAC) unit [26-28]. A proposal to improve the performance of the MAC unit using the modified Radix-4 boot multiplier method was presented in [29]. Lahari et al. [30], approximated various adders in truncated MAC units have been discussed to achieve less area and efficient performance of the FIR filter. Masadeh et al. [31] proposed a MAC unit based on input awareness to resist errors and save energy. According to Ahish et al. [32], the multiplication part of the MAC unit has been implemented based on the "Partial Product Reduction Block (PPRB)" to achieve efficient throughput, latency, and power consumption.

Application-specific- Integrated- Circuits (ASICs) and Field-Programmed Gate Arrays (FPGA) devices developed and built many system designs and algorithms. These devices, especially FPGAs, have the features of speed, parallel processing, and flexible utilization, which have inspired researchers and specialists to increase

the efficiency of systems [33-40]. This work uses an FPGA device to implement a symmetric FIR filter based on single MAC and dual-port RAM. Using this technique, a higher order FIR filter can be achieved with a considerable decrease in hardware size thanks to the FPGA's fast processing abilities.

The rest of the paper is structured as follows: Section 2 offers an overview of the core functions of FIR filters, detailing their common structures and suggesting design methods and implementation approaches. Section 3 features a performance comparison between the proposed FIR filter and a standard filter, derived from simulation outcomes. Section 4 investigates the real-time performance of the FIR filter using HIL co-simulation. The outcomes of the discussions are detailed in Section 5, and the conclusions derived from these results are highlighted in Section 6.

2. Proposed Filter Design

This section delves into the FIR filter, explaining its operational principles, various construction methods, and the advantages of each type. Furthermore, it outlines the design of the proposed filter using the unconstrained Pth norm algorithm to achieve specific frequency response targets. Additionally, it provides an in-depth examination of the FPGA modelling of the filter and discusses the proposed MAC technique for its implementation.

2.1. FIR filter

The following difference equation can express the mathematical model of the digital FIR filter [1]:

$$y(n) = \sum_{i=0}^{N-1} h(i)x(n-i) \quad (1)$$

where y(n) is the output signal at time instant n, x(n) is the input signal at time instant n, h(i) is the ith sample of the filter impulse response, N is the filter length.

Eq. (1) represents the convolution of the input signal x(n) with the filter impulse response h_i s' to produce the output signal y(n). The z-transform of Eq. (1) determines the transfer function of the FIR filter [1].

$$H(z) = \sum_{n=0}^{N-1} h(n)z^{-n} \quad (2)$$

FIR filters can be implemented in various structures. Figure 1 shows two commonly used structures, the direct and transposed forms. The direct form structure of the FIR filter is the direct realization of Eq. (2), while the realization of the transpose form is performed according to the transposition theorem [4]. Although the choice between an FIR filter's transpose and direct form rest on the specific application necessities and constraints, the transpose form offers several advantages over the direct form in terms of numerical stability, computational complexity, implementation flexibility, and frequency response characteristics [4].

The implementation complexity of the FIR filter increases with the increase in its order and required precision. Therefore, with a significant reduction in computational complexity and improved performance, a symmetric FIR filter has become the most popular filter structure. It is achieved by designing it such that the impulse response is symmetric around its midpoint, i.e., h_i=h_(N-i-1). Therefore, Eq. (2) can be modified to the following form [8]:

$$H(z) = \begin{cases} \sum_{k=0}^{\frac{N}{2}} h(k)\left[z^{-k} + z^{-(N-1-k)}\right] & for\ N\ even \\ \sum_{k=0}^{\frac{N-1}{2}} h(k)\left[z^{-k} + z^{-(N-1-k)}\right] + h(\frac{N}{2})z^{-\frac{N}{2}} & for\ N\ odd \end{cases} \quad (3)$$

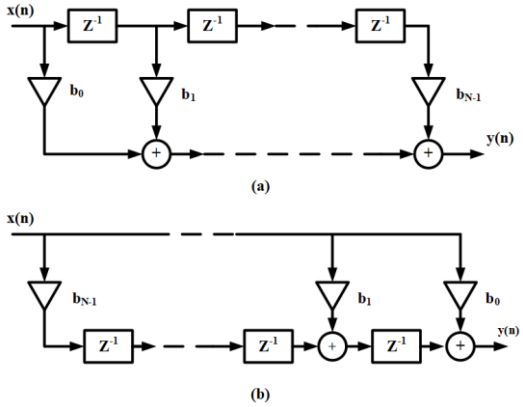

Fig. 1. FIR filter common structures of (a) direct form, (b) transpose form.

Moreover, symmetric FIR filters have a linear phase response, making them useful for applications such as audio processing, where preserving the signal's phase is important. Figure 2 shows the symmetric implementation of the FIR filter. The symmetric direct form structure is shown in Fig. 2(a), while the symmetric transpose structure is shown in Fig. 2(b).

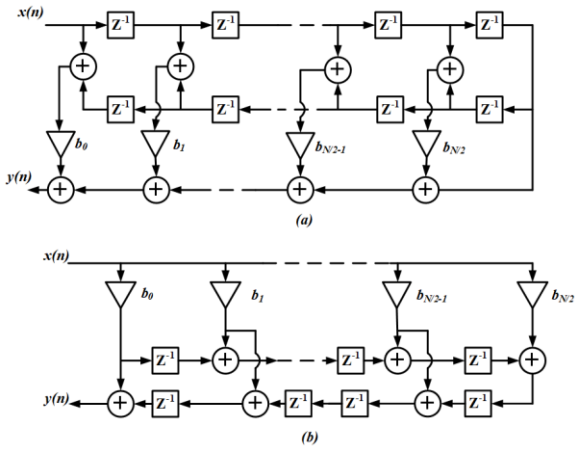

Fig. 2. Symmetric structures of FIR filter (a) direct form, (b) transpose form [8].

Employing a linear phase FIR filter reduces the multipliers count by almost 50% while the number of adders is the same. As the multiplier dominates the hardware area the most, the symmetric structure represents the reduced structure of the FIR filter.

2.2. Design of FIR lowpass filter

Digital FIR filters are the most used due to their natural stability and the presence of large design algorithms. One of the most important methods used in designing filters is the methods that depend on optimization algorithms. Among these optimization algorithms, the unconstrained least-Pth norm optimization algorithm is employed in this paper to extract the filter coefficients. In this algorithm, the design is created based on the complex approximation with the following equation [12]:

$$\|E\|_P = \left(\sum_{i=0}^{L-1}\left(w_i\left|H_i(e^{j\Omega}) - D_i(e^{j\Omega})\right|\right)^P\right)^{\frac{1}{P}} \quad (4)$$

where, $w_i$is a positive weight at ith frequency point, $H_i(e^{j\Omega}) = H(z)|_{z=e^{j\Omega}}$is the actual frequency response of the filter, $D_i(e^{j\Omega})$ is the desired frequency response, L is the filter length, P is the Pth norm, and $\|E\|_P$ is the Pth norm of the cost function E to be minimized regarding FIR coefficients.

2.3. MAC-based FIR filter

Basic N-Tap FIR filter implementation principles are represented by the structures in Fig. 2, where each tap has a multiplier, adder, and delay units. Therefore, increasing the number of taps will increase the hardware implementation's complexity. Several techniques have been presented in the literature that aim to reduce the complexities of FIR filter construction. Although the choice of technique depends on the desired characteristics, hardware resources, and application requirements, The single MAC approach for FIR filters provides advantages like cost-effectiveness, lower power consumption, and decreased hardware needs. It is perfect for real-time applications with constrained resources, guaranteeing consistent latency and simplifying the architecture. This section will present an explanation of employing a single MAC block to perform a serial MAC operation of an Nth-order FIR filter [13].

The fundamental strategy of the FIR filter implementation grounded on a single MAC block is depicted in Fig. 3. Two register blocks are used in this architecture. While the second register stores the filter coefficients, the first collects input signal samples. At the beginning of the block, two input values are multiplied, and the resulting product is then added to an accumulator register. This accumulator enables the MAC block to efficiently execute operations such as dot products within a single instruction cycle by maintaining the running total of multiple products.

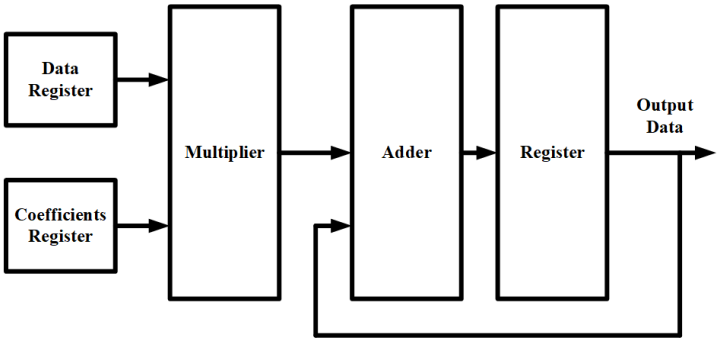

Fig. 3. Single MAC technique-based FIR filter implementation.

2.4. Modelling based XSG

The Xilinx system generator is a high-level design tool that works well for designing unique DSP data routes in the FPGAs. The generator provides a high definition of FPGA circuits; it can also be used to create rapid and small hand-based prototypes [5, 33, 34]. Figure 4 shows the system generator model of FIR filter implementation in MAC. The delay block of the System Builder model operates at data input speed, while the memory and MAC blocks operate at N-times this rate, using subsampling blocks to reduce the data throughput rate

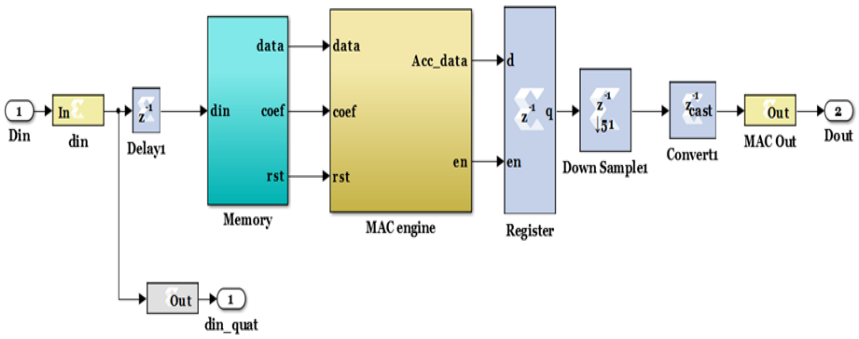

Fig. 4. XSG-based FIR filter implementation.

In this instance, the memory block is set up as addressable dual-port Random access Memory (RAM) with reconfigurable depth, which allows the memory's size or depth to be changed per the filter length. Addressable dual port RAM architecture enables simultaneous access to the memory by two separate devices or processors. Two read or write operations can be carried out concurrently or separately thanks to the two different sets of address and data lines that are present. Figure 5 shows the configuration of the addressable dual-port RAM to govern the exchange of the signal samples and the filter coefficients.

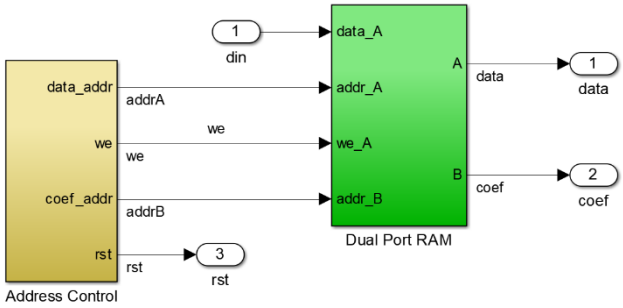

Fig. 5. XSG-based addressable dual-port RAM.

MAC engines are used in a flexible FIR filter architecture that fits well on FPGA resources. Figure 6 depicts a single-MAC FIR filter model created using the System Generator. This incredibly small architecture makes sense for applications requiring low throughput or when N is tiny. The MAC engine consists of balance

and block accumulators. The multiplier calculates the product of the filter and the sample in the data buffer, and the accumulator calculates the running count of these results. Factors can be used in logical models or as multipliers. To avoid looping at the end of each product calculation, the accumulator is programmed to reset to its current value during reset. The scratchpad stores the output of the MAC engine before rebooting. The output of the capture register is downsampled so that the filter output equals the input MAC engines are used in a flexible FIR filter architecture that fits well on FPGA resources.

Figure 6 depicts a single-MAC FIR filter model created using the System Generator. This incredibly small architecture makes sense for applications requiring low throughput or when N is tiny. The MAC engine consists of balance and block accumulators. The multiplier calculates the product of the filter and the sample in the data buffer, and the accumulator calculates the running count of these results. Factors can be used in logical models or as multipliers. To avoid looping at the end of each product calculation, the accumulator is programmed to reset to its current value during reset. The scratchpad stores the output of the MAC engine before rebooting. The output of the capture register is downsampled so that the filter output equals the input value.

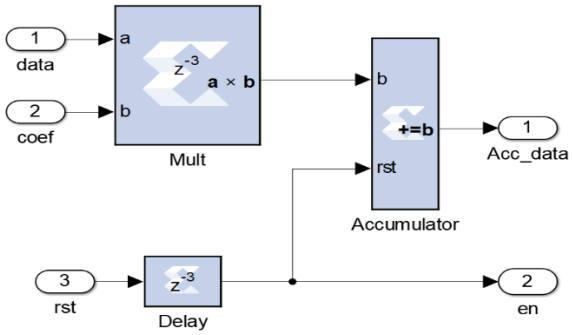

Fig. 6. XSG-based single-MAC unit.

3. Simulation Results

Verifying a designed FIR filter's efficacy ensures it runs accurately and meets the desired requirements. The FIR filter is normally designed by the intended requirements as the initial step before the validation process. The designed FIR filter in this paper has the specifications depicted in Table 1.

Table 1. FIR filter specifications.

Filter Type	Lowpass
Design method	Least Pth-Norm
Filter order	51
Density factor	20
Pth-Norm	128
Sampling frequency	44.1KHz
Passband edge frequency	3KHz
Stopband edge frequency	5KHz

The MATLAB signal processing Toolbox's FDA Tool, which features a graphical user interface (GUI), has been used to extract the filter coefficients according to the requirements. The filter coefficients will be exported to the MATLAB workspace after the filter has been designed using the FDATool in MATLAB. These coefficients represent the transfer function's numerical values, which entirely describe the filter's behaviour. Coefficients can be used in the Simulink model with the Xilinx System Generator (XSG) for FPGA implementation once they are in the workspace. Moreover, the designed filter has been compared with the ideal direct form II transpose FIR filter built in the MATLAB/ Simulink package. This comparison assists in validating the accuracy of the filter design and makes sure the filter satisfies the required criteria. Figure 7 depicts the system configuration for comparing the reference and proposed FIR filters based on XSG. The source signal was random with a uniform distribution and peak-to-peak value of ±1.99.

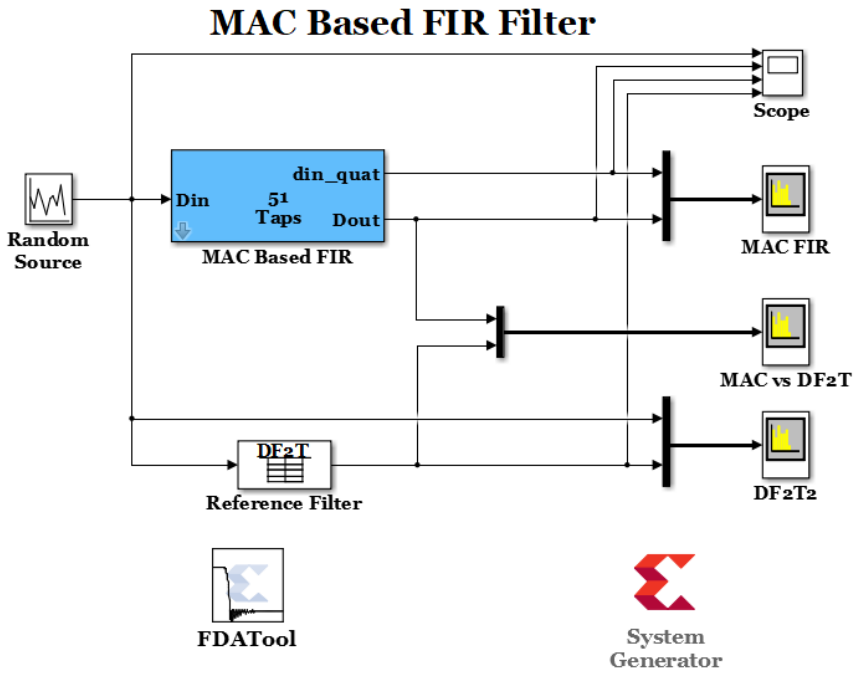

Fig. 7. System configuration for comparing the reference and proposed FIR filters based on XSG.

Simulation results are shown in Fig. 8, and the frequency response of the reference FIR filter versus the source signal is shown in Fig. 8(a). In Fig. 8(b), the frequency response of the designed FIR filter about the source signal is displayed. Fig. 8(c) compares the developed FIR filter's frequency response to that of the reference FIR filter.

The simulation results are displayed in Fig. 9, along with a time domain comparison of the performance of the reference and proposed filters. Figure 9(a) shows the noisy source signal and has a peak-to-peak amplitude of 1.99. It is uniformly distributed random noise. The outcomes of processing this signal using

the reference FIR and the FIR filters constructed according to the specifications are displayed in Figs. 9(b) and 9(c), respectively.

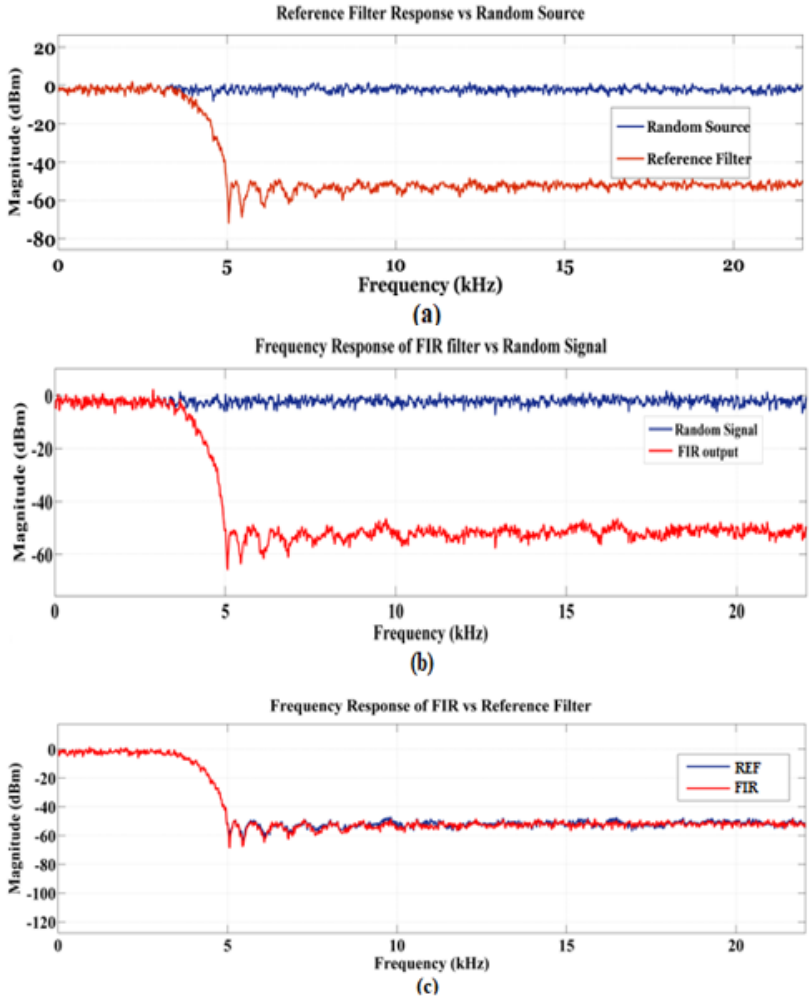

Fig. 8. Frequency response: (a) reference filter, (b) designed filter, (c) reference vs. designed.

4. HIL Co-Simulation Result

The filter operation was validated in real-time using Hardware-in-Loop (HIL) co-simulation. In this approach, the target hardware was a Xilinx Spartan6 FPGA device (XC6SLX45T) housed in the SP605 Evaluation Kit, which interfaced with a PC running MATLAB/Simulink. The FPGA implemented the filter, and Simulink input/output blocks were connected to it. The input and output ports were defined by a JTAG Co-Sim block, created with Xilinx System Generator (XSG). The target device was connected to the computer via a USB-type B connector to facilitate real-time co-simulation as shown in Fig. 10. Table 2 presents a summary of the resource utilization on the target device for implementing the proposed filter.

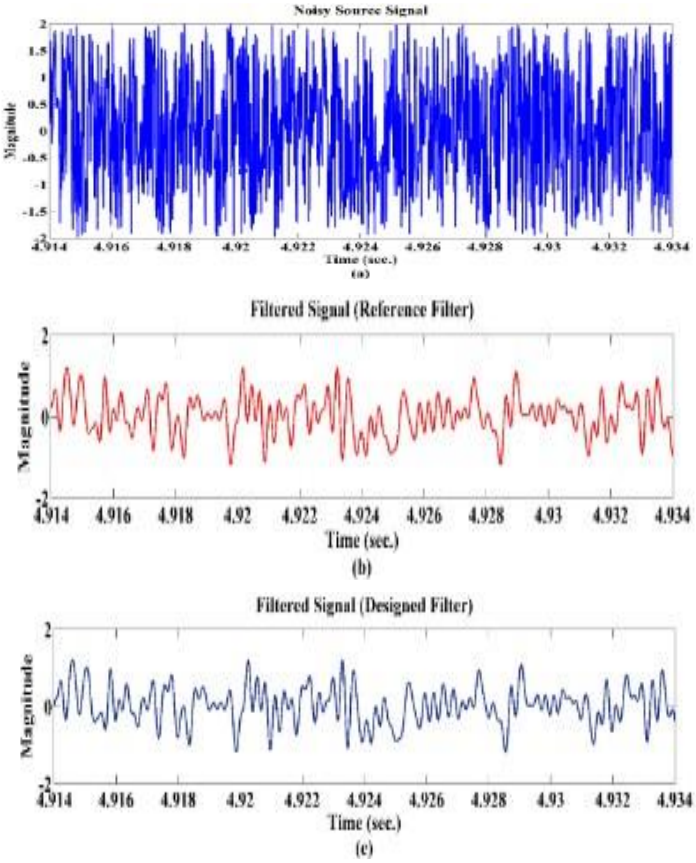

Fig. 9. Time response: (a) noisy input, (b) reference filter, (c) designed filter.

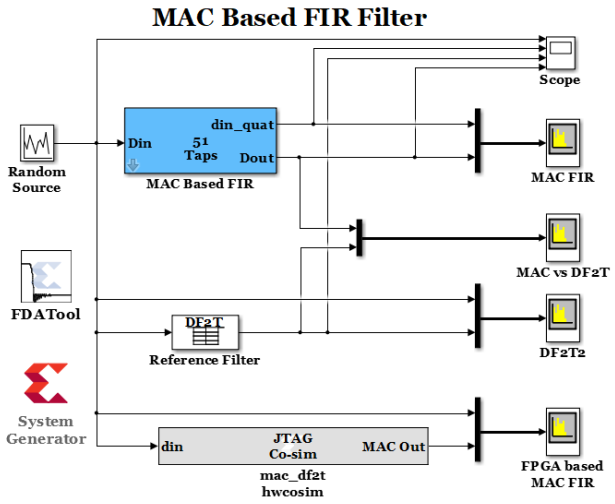

Fig. 10. HIL Co-simulation of the designed filter.

Table 2. Device utilization summary.

No. of Taps	**51**
Size of input data	10-bit, 8-bit binary point.
Size of filter coefficients	12-bit, 12-bit binary point.
No. of Slices Registers	129 out of 54576.
No. of Slices LUTs	100 out of 33696.
Number of LUT-FFs pair	181
No. of Block RAM/FIFO	1 out of 116.
No. of DSP48A1S	1 out of 58.

The result of the (HIL) Co-simulation is presented in Fig. 11, which represents the frequency response of the designed FIR filter in real-time.

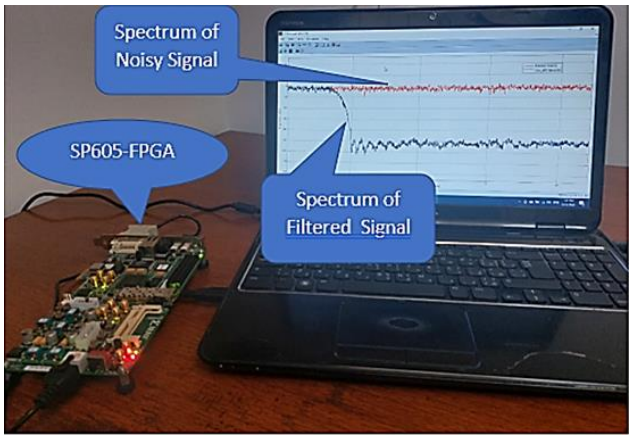

Fig. 11. Real-time frequency response of the designed filter.

5. Results and Discussion

The proposed lowpass FIR filter performance was examined in two parts. The first part was the simulation in which its frequency and time domain performance are compared with the reference lowpass FIR filter, as shown in Figs. 8 and 9. From the frequency response point of view, simulation results in Fig. 8 show that the performance of the proposed filter matches the requirements listed in Table 1 (i.e., the cutoff frequency and the stop band edge). In addition, the results also showed that the performance of the designed filter was identical to the reference filter performance. In the time domain, the results depicted in Fig. 9 which showed that the proposed filter's time response matches the reference filter's response.

In the second part, the performance of the proposed FIR filter was validated in real-time using an FPGA board in the HIL Co-Simulation process, as illustrated in Fig. 10. The experimental result in Fig. 11 confirms that the desired frequency response was achieved. Table 3 compares the dynamic power consumption and device utilization between the proposed FIR filter design and the current one for XC6SLX45T.

Table 3. Comparison of device usage and power consumption.

Method	Proposed	[8]
No. of Taps	51	50
Slices	229	2945
Filp-Flops	181	3132
Block RAM/FIFO	1	-
DSP48A1S	1	-
Dynamic power consumption (mW)	10	37.8245
Leakage power (mW)	36	0.0074

According to Table 3, the proposed FIR filter design demonstrates reduced device usage and dynamic power consumption compared to the design outlined in [8]. The reduction in dynamic power consumption and increased leakage power indicate that the proposed approach effectively minimizes device switching variation.

A future perspective of this work is to introduce artificial intelligence, like deep learning, to synthesize the FIR or improve the proposed filter's characteristics. Also, one can make a comparison study of this work to other embedded system technologies [41-50]. The FIRs are very instructive in control applications, which require filtered or smoothed feedback signals using state-feedback or observer-based control techniques. In this sense, the FIR is crucial in providing noise-free signals to avoid processing false signals by different controllers [51-65].

6. Conclusions

In this study symmetric lowpass FIR filter architecture is created utilizing the Xilinx system generator and implemented using FPGA platforms. The FDA Tool-based unconstrained Pth-norm optimization approach has determined the FIR filter coefficients based on the desired specifications. A single MAC unit and addressable dual-pot RAM implement the FIR filter architecture. It is promising that the proposed architecture has notably decreased the number of logic components, and power consumption (as indicated in tables 2 and 3) thereby enhancing scalability and efficiency. The adaptable nature of this architecture allows for the straightforward implementation of FIR (Finite Impulse Response) filters of any order. However, implementing higher-order filters demands additional memory, posing a challenge to scalability. This increased memory requirement can create issues, particularly in systems with limited resources. The reconfigurability, speed, and parallel processing abilities of FPGAs enhance the proposed architecture's ability to manage the expansion of filter orders.

References

1. Mittal, A.; Nandi, A.; and Yadav, D. (2017). Comparative study of 16-order FIR filter design using different multiplication techniques. *IET Circuits, Devices & Systems*, 11 (3), 196-200.
2. Trimble, M.B.; and Chilveri (2017). A review: FIR filter implementation. *Proceedings of the 2nd IEEE International Conference On Recent Trends In Electronics Information & Communication Technology* (*RTEICT*, 2017), Bengaluru, India, 137-141.
3. Sudharsan, R.R. (2019). Synthesis of FIR filter using ADC-DAC: A FPGA implementation. *Proceedings of the IEEE International Conference on Clean*

Energy and Energy Efficient Electronics Circuit for Sustainable Development (*INCCES*, 2019), Krishnankoil, India, 1-3.

4. Khurshid, B.; Manzoor, M.; Muzaffar, K.; Wani, B.; and Mushtaq, B. (2019). Achieving performance speed-up in FPGA based FIR filters using DSP macroblocks. *Proceedings of the 6th International Conference on Signal Processing and Integrated Networks* (*SPIN*, 2019), Noida, India. 190-194.
5. Hassan, R.F. (2021). Performance investigation of digital lowpass IIR filter based on different platforms. *International Journal of Electrical and Computer Engineering Systems*, 12(2), 105-111.
6. Hatai, I.; Chakrabarti, I.; and Banerjee, S. (2015). An efficient constant multiplier architecture based on vertical-horizontal binary common sub-expression elimination algorithm for reconfigurable FIR filter synthesis. *IEEE Transactions on Circuits and Systems I*: *Regular Papers*, 62(4), 1071-1080.
7. Reddy, K.S.; Madhavan, S.; Falkowski-Gilski, P.; Divakarachari, P.B.; and Mathiyalagan, A. (2022). Efficient FPGA implementation of an RFIR filter using the APC–OMS technique with WTM for high-throughput signal processing. *Electronics*, 11(19), 3118.
8. Touil, L.; Hamdi, A.; Gassoumi, I.; and Mtibaa, A. (2020). Design of low-power structural FIR filter using data-driven clock gating and multibit flip-flops. *Journal of Electrical and Computer Engineering*, 2020(1), 8108591.
9. Patali, P.; and Kassim, S.T. (2019). High throughput FIR filter architectures using retiming and modified CSLA based adders. *IET Circuits, Devices & Systems*, 13(7), 1007-1017.
10. Das, R.; Guha, A.; and Bhattacharya, A. (2016). FPGA based higher order fir filter using XILINX system generator. *Proceedings of the International Conference on Signal Processing, Communication, Power and Embedded System* (*SCOPES*, 2016), Paralakhemundi, India, 111-115.
11. Rengaprakash, S. et al. (2017). FPGA implementation of fast running FIR filters. *Proceedings of the International Conference on Wireless Communications, Signal Processing and Networking* (*WiSPNET*, 2017), Chennai, India, 1282-1286.
12. Lu, W.-S. (2002). Minimax design of nonlinear-phase FIR filters: a least-pth approach. *Proceedings of* 2002 *IEEE International Symposium on Circuits and Systems*, Phoenix-Scottsdale, AZ, USA, I-I.
13. Kumar, N.U.; Subbalakshmi, U.; Priya, B.S.; and Sindhuri, K.B. (2018). VLSI implementation of FIR filter using different addition and multiplication techniques. *Proceeding of the International Conference on Soft Computing Systems* (*ICSCS* 2018), Kollam, India, 483-490.
14. Chauhan, A.; and Kumar, P.S. (2018). Implementation of FIR filter & Mac unit by using neural networks in FPGA. *Proceedings of the International Conference on Advances in Computing, Communications and Informatics* (*ICACCI*, 2018), Bangalore, India, 2496-2501.
15. Maamoun, M. et al. (2021). Efficient FPGA based architecture for high-order FIR filtering using simultaneous DSP and LUT reduced utilization. *IET Circuits, Devices & Systems*, 15(5), 475-484.

16. Park, J.; Muhammad, K.; and Roy, K. (2003). High-Performance FIR Filter Design Based on Sharing Multiplication. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 11(2), 244-253.
17. Kaur, R.; Malhotra, R.; and Deb, S. (2015). MAC-based FIR filter: A novel approach for low-power real-time de-noising of ECG signals. *Proceedings of the* 2015 19*th International Symposium on VLSI Design and Test*, Ahmedabad, India, 1-5.
18. Meher, P.K.; Chandrasekaran, S.; and Amira, A. (2008). FPGA realization of FIR filters by efficient and flexible systolization using distributed arithmetic. *IEEE Transactions on Signal Processing*, 56(7), 3009-3017.
19. Chandra, A.; and Chattopadhyay, S. (2016). Design of hardware efficient FIR filter: A review of the state-of-the-art approaches. *Engineering Science and Technology, an International Journal*, 19(1), 212-226.
20. Nassralla, M.H.; Akl, N.; and Dawy, Z. (2021). A clustering-based approach for designing low complexity FIR filters. *IEEE Signal Processing Letters*, 28, 299-303.
21. Hartley, R.I. (1996). Subexpression sharing in filters using canonic signed digit multipliers. *IEEE Transactions on Circuits and Systems* II: *Analog and Digital Signal Processing*, 43(10), 677-688.
22. Pasko, R.; Schaumont, P.; Derudder, V.; Vernalde, S.; and Durackova, D. (1999). A new algorithm for elimination of common subexpressions. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 18(1), 58-68.
23. Martínez-Peiró, M.; Boemo, E.I.; and Wanhammar, L. (2002). Design of high-speed multiplierless filters using a nonrecursive signed common subexpression algorithm. *IEEE Transactions on Circuits and Systems* II: *Analog and Digital Signal Processing*, 49(3), 196-203.
24. Mahesh, R.; and Vinod, A.P. (2007). A new common subexpression elimination algorithm for realizing low-complexity higher order digital filters. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 27(2), 217-229.
25. Liacha, A.; Oudjida, A.K.; Ferguene, F.; Bakiri, M.; and Berrandjia, M.L. (2018). Design of high-speed, low-power, and area-efficient FIR filters. *IET Circuits, Devices* & *Systems*, 12(1), 1-11.
26. Kumar, V.N.; Nalluri, K.R.; and Lakshminarayanan, G. (2015). Design of area and power efficient digital FIR filter using modified MAC unit. *Proceedings of the 2nd International Conference on Electronics and Communication Systems* (*ICECS*, 2015), Coimbatore, India, 884-887.
27. Lyakhov, P.; Valueva, M.; Valuev, G.; and Nagornov, N. (2020). A method of increasing digital filter performance based on truncated multiply-accumulate units. *Applied Science*, 10(24), 9052.
28. Rakesh, H.M.; and Sunitha, G.S. (2020). Design and implementation of novel 32-bit MAC unit for DSP applications. *Proceedings of the International Conference for Emerging Technology* (*INCET*, 2020), Belgaum, India, 1-6.
29. Patil, P.A.; and Kulkarni, C. (2018). Multiply accumulate unit using radix-4 booth encoding. *Proceedings of the Second International Conference on*

Intelligent Computing and Control Systems (*ICICCS*, 2018), Madurai, India, 1076-1080.

30. Lahari, P.L.; Bharathi, M.; and Shirur, Y.J. (2020). An efficient truncated MAC using approximate adders for image and video processing applications. *Proceedings of the* 4*th International Conference on Trends in Electronics and Informatics* (*ICOEI*, 2020), Tirunelveli, India, 1039-1043.
31. Masadeh, M.; Hasan, O.; and Tahar, S. (2019). Input-conscious approximate multiply-accumulate (MAC) unit for energy-efficiency. *IEEE Access*, 7, 147129-147142.
32. Ahish, S.; Kumar, Y.B.N.; Sharma, D.; and Vasantha, M.H. (2015). Design of high-performance multiply-accumulate computation unit. *Proceedings of* 2015 *IEEE International Advance Computing Conference* (*IACC*, 2015), Banglore, India, 915-918.
33. Kasim, M.Q. et al. (2021). Control algorithm of five-level asymmetric stacked converter based on xilinx system generator. *Proceedings of the IEEE* 9*th Conference on Systems, Process and Control* (*ICSPC* 2021), Malacca, Malaysia, 174-179.
34. Hassan, R.F.; Ajel, A.R.; Abbas, S.J.; and Humaidi, A.J. (2022). FPGA based hil co-simulation of 2DOF-PID controller tuned by PSO optimization algorithm. *ICIC Express Letters*, 16(12), 1269-1278.
35. Dalloo, A.M.; Humaidi, A.J.; Mhdawi, A.K.A.; and Al-Raweshidy, H. (2024). Low-power and low-latency hardware implementation of approximate hyperbolic and exponential functions for embedded system applications. *IEEE Access*, 12, 24151-24163.
36. Humaidi, A.J.; Kadhim, T.M.; Hasan, S.; Ibraheem, I.K.; and Azar, A.T. (2020). A generic izhikevich-modelled FPGA-realized architecture: A case study of printed English letter recognition. *Proceedings of the* 24*th International Conference on System Theory, Control and Computing* (*ICSTCC*, 2020), Sinaia, Romania, 825-830.
37. Dalloo, A.M.; and Humaidi, A.J. (2025). Optimizing machine learning models with data-level approximate computing: The role of diverse sampling, precision scaling, quantization and feature selection strategies. *Results in Engineering*, 24, 103451.
38. Humaidi, A.J.; and Kadhim, T.M. (2028). Spiking versus traditional neural networks for character recognition on FPGA platform. *Journal of Telecommunication, Electronic and Computer Engineering*, 10(3), 109-115.
39. Dalloo, A.M.; Humaidi, A.J.; Mhdawi, A.K.A.; and Al-Raweshidy, H. (2024). Approximate computing: concepts, architectures, challenges, applications, and future directions. *IEEE Access*, 12, 146022-146088.
40. Al-Kaysi, A.M.; Ali, S.M.; Mahmood, N.S.; and Humaidi, A.J. (2024). Design of ECG monitoring system for cardiovascular patients using LABVIEW**.** *Journal of Engineering Science and Technology*, 19(4), 1189-1205.
41. Nasser, A.R.; Hasan, A.M.; and Humaidi, A.J. (2024). DL-AMDet: Deep learning-based malware detector for Android. *Intelligent Systems with Applications*, 21(200318).

42. Korial, A.E.; Gorial, I.I.; and Humaidi, A.J. (2024). An improved ensemble-based cardiovascular disease detection system with chi-square feature selection. *Computers*, 13(6), 126.
43. Abed, R.A.; Hamza, E.K.; and Humaidi, A.J. (2024). A modified CNN-IDS model for enhancing the efficacy of intrusion detection system. *Measurement*: *Sensors*, 35, 101299.
44. Hamza, E.K.; Salman, K.D.; and Jaafar, S.N. (2023). *Wireless sensor network for robot navigation*. In Azar, A.T., Kasim Ibraheem, I., Jaleel Humaidi, A. (Eds.), *Mobile Robot: Motion Control and Path Planning. Studies in Computational Intelligence*. Springer, Cham, 643-670.
45. Alhasan, R.A.; and Hamza, E.K. (2023). A novel CNN model with dimensionality reduction for WSN intrusion detection. *Revue d'Intelligence Artificielle*, 37(5), 1121-1131.
46. Hasan, R.A.A.; and Hamza, E.K. (2023). An improved intrusion detection system using machine learning with singular value decomposition and principal component analysis. *International Journal of Intelligent Engineering and Systems*, 16(4), 25-38.
47. Sabbar, E.H. et al. (2023). Effects of Ag content on microstructure evolution, intermetallic compound (IMC) and mechanical behavior of SAC solder joints. *Microelectronics Reliability*, 147, 115103.
48. Al-Janabi, S.F.; and Obaid, A.K. (2012). Development of certificate authority services for web applications. *Proceedings of the* 2012 *International Conference on Future Communication Networks*, Baghdad, Iraq, 135-140.
49. Obaid, A.K.; Abdel-Qader, I.; and Mickus, M. (2018). Automatic food-intake monitoring system for persons living with Alzheimer's-vision-based embedded system. *Proceedings of the* 9*th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference* (*UEMCON*), New York, NY, USA, 580-584.
50. Abed, A.; Mansoor, R.; and Kais, A. (2022). Multi-patterns-based peak to average power ratio reduction in OFDM systems. *Journal of Communications Technology and Electronics*, 67, 834-842.
51. Hameed, A.H.; Al-Samarraie, S.A.; Humaidi, A.J.; and Saeed, N. (2024). Backstepping-based quasi-sliding mode control and observation for electric vehicle systems: a solution to unmatched load and road perturbations. *World Electric Vehicle Journal*, 15(9), 419.
52. Zouari, F.; Ibeas, A.; Boulkroune, A.; and Cao, J. (2024). Finite-time adaptive event-triggered output feedback intelligent control for noninteger order nonstrict feedback systems with asymmetric time-varying pseudo-state constraints and nonsmooth input nonlinearities. *Communications in Nonlinear Science and Numerical Simulation*, 136, 108036.
53. Abbas, S.J.; Husain, S.S.; Al-Wais, S.; and Humaidi, A.J. (2024). Adaptive integral sliding mode controller (SMC) design for vehicle steer-by-wire system. *SAE International Journal of Vehicle Dynamics, Stability, and NVH*, 8(3), 383-396.
54. Zouari, F.; Saad, K.B.; and Benrejeb, M. (2012). Robust neural adaptive control for a class of uncertain nonlinear complex dynamical multivariable systems. *International Review on Modelling and Simulations*, 5(5), 2075-2103.

55. Hameed, A.H.; Al-Samarraie, S.A.; and Humaidi, A.J. (2024). Ultimate bounded observer-based control of electrical vehicle driven by DC motor system with unmatched load torque. *Advances in Mechanical Engineering*, 16(10), 16878132241273549.
56. Zouari, F.; Saad, K.B.; and Benrejeb, M. (2013). Adaptive backstepping control for a single-link flexible robot manipulator driven DC motor. *Proceedings of the International Conference on Control, Decision and Information Technologies* (*CoDIT*, 2013), Hammamet, Tunisia, 864-871.
57. Rigatos, G.; Abbaszadeh, M.; Siano, P.; Al-Numay, M.; and Zouari, F. (2024). A nonlinear optimal control approach for bacterial infections under antibiotics resistance. *Journal of Systems Science and Complexity*, 37, 2293-2317.
58. Hameed, A.H.; Al-Samarraie, S.A.; and Humaidi, A.J. (2024). Reduced ultimate-bound of tracking error convergence for ETV system with unknown upper-bound mismatched perturbation. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 09544070241272879.
59. Zouari, F.; Saad, K.B.; and Benrejeb, M. (2013). Adaptive backstepping control for a class of uncertain single input single output nonlinear systems. *Proceedings of* 2013 10*th International Multi-Conference on Systems, Signals & Devices* (*SSD*13), Hammamet, Tunisia, 1-6.
60. Humaidi, A.J.; and Hameed, M.R. (2017). Design and performance investigation of block-backstepping algorithms for ball and arc system. *Proceedings of* 2017 *IEEE International Conference on Power, Control, Signals and Instrumentation Engineering* (*ICPCSI*), Chennai, India, 325-332.
61. Boulkroune, A.; Hamel, S.; Zouari, F.; Boukabou, A.; and Ibeas, A. (2017). Output-feedback controller-based projective lag-synchronization of uncertain chaotic systems in the presence of input nonlinearities. *Mathematical Problems in Engineering*, 2017(1), 8045803.
62. Ajel, A.R.; Humaidi, A.J.; Ibraheem, I.K.; and Azar, A.T. (2021). Robust model reference adaptive control for tail-sitter VTOL aircraft. *Actuators*, 10(7), 162.
63. Rigatos, G. et al. (2023). Nonlinear optimal control for a gas compressor driven by an induction motor. *Results in Control and Optimization*, 11(100226).
64. Ibeas, A.; Esmaeili, A.; Herrera, J.; and Zouari, F. (2016). Discrete-time observer-based state feedback control of heart rate during treadmill exercise. *Proceedings of* 20*th International Conference on System Theory, Control and Computing* (*ICSTCC*), Sinaia, Romania, 1-6.
65. Rigatos, G.; Siano, P.; Wira, P.; Abbaszadeh, M.; and Zouari, F. (2018). Nonlinear H-infinity control for optimization of the functioning of mining products mills. *Proceedings of IECON* 2018 - 44*th Annual Conference of the IEEE Industrial Electronics Society*, Washington, DC, USA, 2367-2372 .