International Journal of Creative and Open Research in Engineering and Management
ISSN: **3108-1754 (Online)**
Volume 02 Issue 05 May-2026 | Impact Factor: 3.5

FPGA Oriented RRPG Based FIR Filter with LMS Adaptive Extension to Reduce Error Performance

Dr. N. Suneetha
Associate Professor, Department of ECE
Sir C R Reddy College of Engineering(A)
Eluru, West Godavari Dt-534007, India
suneethavelamati@gmail.com

M. Charan Naga Sabareesh
Department of Electronics and Communication Engineering
Sir C R Reddy College of Engineering(A)
Eluru, West Godavari Dt-534007, India
charanmurala100@gmail.com

K. Bhavya Sri
Department of Electronics and Communication Engineering
Sir C R Reddy College of Engineering(A)
Eluru, West Godavari Dt-534007, India
bhavyasrikadali8@gmail.com

K . Rahul
Department of Electronics and Communication Engineering
Sir C R Reddy College of Engineering(A)
Eluru, West Godavari Dt-534007, India
kotarahul62@gmail.com

M. Karthik Chowdary
Department of Electronics and Communication Engineering
Sir C R Reddy College of Engineering(A)
Eluru, West Godavari Dt-534007, India
karthikchowdarymaddineni@gmail.com

P. Tejeswar Raj
Department of Electronics and Communication Engineering
Sir C R Reddy College of Engineering(A)
Eluru, West Godavari Dt-534007, India
tejroyal131@gmail.com

Abstract--

Finite Impulse Response (FIR) filters are required in the digital signal processing systems like communication systems and analysis of signals. Traditional multiplier-based realizations create a high level of complexity in the hardware and Distributed Arithmetic (DA) lowers the count of multipliers at the cost of a memory overhead. New architectures like Reconfigurable Partial Product Generator (RRPG) based FIR filters are more efficient in the use of hardware because they use fewer resources. The FIR filter is implemented as the RRPG based baseline architecture in this work and it is subsequently improved with Least Mean Square (LMS) adaptive algorithm. The LMS algorithm compares the desired and actual output and updates filter coefficients in an iterative manner to minimize the error between Filtering, Error Reduction. them thus further enhancing filtering accuracy. The suggested RRPG+LMS architecture is tested using MATLAB simulation and implemented in Verilog HDL with Xilinx Vivado. The experimental findings have shown that when the LMS is integrated, the mean square error is greatly minimized compared to the RRPG-based FIR filter operating alone whereas maintaining efficient hardware performance. The fact that the results of the MATLAB and the hardware are close to each other proves the accuracy and usefulness of the given approach to real-time FPGA-based systems.

Keywords: FIR Filter, RRPG, LMS Algorithm, FPGA, Adaptive

1.INTRODUCTION

FIR filters have become popular in digital signal processing systems like communication systems and signal analysis because they are stable and have a linear phase response. But traditional multiplier-based implementations cause high hardware complexity and more resource consumption on FPGA platforms.[1]

Distributed Arithmetic (DA) simplifies this by using Look-Up Tables (LUTs) instead of multipliers and shift-adds. Despite the fact that DA enhances the efficiency of hardware, the size of the LUT grows exponentially with the order of the filter, and as a result, the memory demands become high. To solve this problem, Reconfigurable Partial Product Generator (RRPG) based FIR architectures are more efficient, as they minimize the memory and the computational complexity.

But the RRPG-based filters are fixed coefficient filters, and they might not be effective in different signal environments. To address this shortcoming, this paper extends the RRPG-based FIR filter by Least Mean Square (LMS) adaptive algorithm which adjusts the filter coefficients through an iterative process to decrease error.[2]

The rest of the paper is structured in the following way. Section 2 includes background and related work, Section 3 is the description of proposed method, Section 4 is results and discussion and finally, Section 5 includes conclusion and future scope.

2.BACKGROUND AND RELATED WORK

Finite Impulse Response (FIR) filters are popular in digital signal processing application because of their stability and linear phase. Different methods of hardware implementation have been created to enhance efficiency and eliminate complexity.

A. Multiplier-Based FIR

The traditional FIR filters are implemented using multipliers and adders to carry out convolution between input samples and filter coefficients. Though the method has accurate results, it has a high hardware complexity and more resources are used with an increase in the order of the filter.

B. Distributed Arithmetic (DA)

Distributed Arithmetic (DA) is a technique, which has no multiplier, instead of performing multiplication operations it uses Look-Up Tables (LUTs) and shift-add operations. This makes hardware more efficient, but the size of the LUT increases exponentially with the filter order, leading to the need of more memory.[3]

C. Optimized RRPG-DA

To address the shortcomings of traditional DA, the proposals have been done on optimized architectures like Reconfigurable Partial Product Generator (RRPG). RRPG eliminates LUT size by storing only some combinations by generating partial products dynamically through a set of shift-and-add operations. This enhances memory performance and computational performance and is therefore applicable in FPGA implementations.

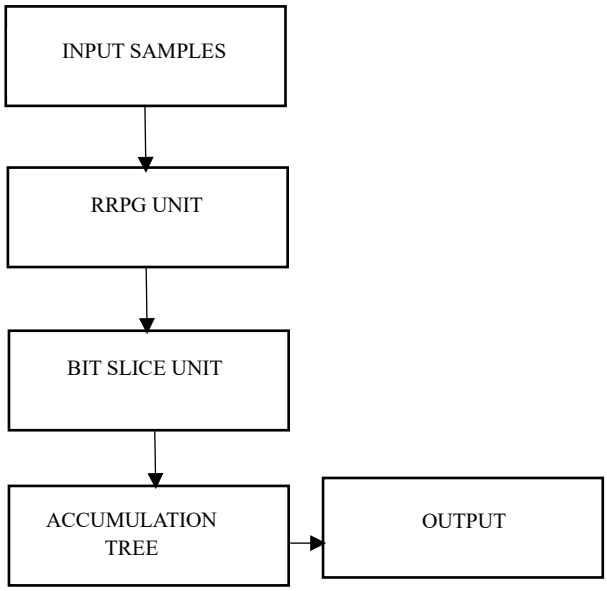

Fig. 1:RRPG-Based FIR Filter Architecture

The FIR filter architecture based on the RRPG is a dynamically generated partial products based on shift-and-add operation to require less memory space and higher computational efficiency than the conventional DA-based implementation.

D. LMS Adaptive Filtering

In practice, the characteristics of a signal change over time, and thus fixed-coefficient filters are less useful. Least Mean Square (LMS) algorithm is also popular in adaptive filtering, in which the filter coefficients are continually optimized to ensure that the difference between the desired and the actual outputs is minimized [4].

E. Research Gap

Even though RRPG enhances the efficiency of FIR filters based on the idea of DA, it has fixed coefficients and is not adaptable in changing conditions of the signal. LMS-based adaptive filtering, on the other hand, is more accurate but is not concerned with hardware optimization. Thus, a combination of RRPG and LMS can be used as an effective and flexible solution to real-time signal processing.[4]

3.RESEARCH METHODOLOGY

In this section we discuss the methodology employed in the development of an adaptive FIR filter that we designed. As it can be seen below, this filter employs a certain architecture optimized through the use of a reconfigurable partial product generator or RRPG. Besides, the least mean square algorithm has been employed in this work for the purpose of reducing error.[5]

A.FIR Filter Model

The FIR filter performs the convolution of the input signal and the coefficient vector. Thus, there is input vector x, coefficient vector h and output vector y and is given by

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

B.RRPG-Based FIR Implementation

The main advantage of using an RRPG architecture lies in the possibility to eliminate look up table and multiplier elements that occupy relatively large areas of chip. This can be done using the shift and addition operations that help produce partial products. There is no necessity to employ large precomputed and stored lookup tables here. Partial products are then segmented to produce an output using a tree-like accumulation structure.[5]

C. LMS Based Adaptive Filtering

In order to increase the accuracy of the filter, we have employed the least mean square algorithm for the purpose of adaptively changing the coefficients. [6]In this respect, it should be pointed out that the error of the FIR filter is the difference between the input vector d and the output vector y. It may be expressed as

$$e(n) = d(n) - y(n) \qquad (2)$$

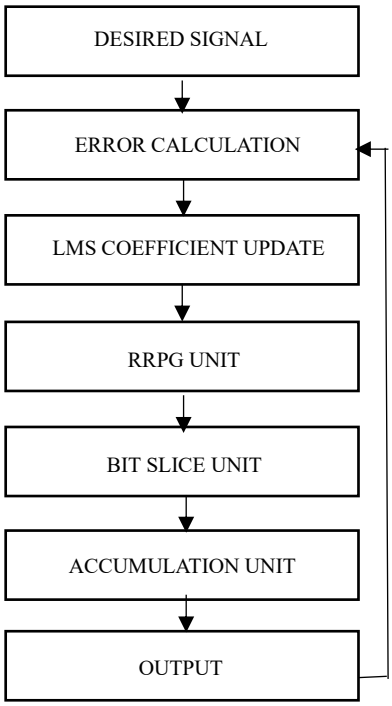

Fig. 2:Proposed RRPG-LMS Based FIR Filter Architecture

Moreover, the update of the coefficients can be given as follows

$$w(n+1) = w(n) + \mu e(n)x(n) \qquad (3)$$

Here the parameter μ represents a certain step size.

Thus, as it can be seen above, the adaptive FIR filter we have developed includes the RRPG-based architecture along with the LMS algorithm. While RRPG deals with calculations, LMS takes care of adaptation.[2]

4. RESULTS AND DISCUSSIONS

Here's a closer look at the performance of the suggested RRPG-driven FIR filter. Accuracy of results comes first, followed by the degree of error reduction. Speed during execution matters just as much. Hardware behavior wraps up the picture.[7]

Not long after starting, we ran the algorithm in MATLAB just to confirm things worked right. With those outcomes double-checked, across came everything into Verilog HDL. Hardware behavior got a look through tests run inside Xilinx Vivado. From beginning to end, one steady input signal guided the process along with fixed filter setups. Every test stayed under these conditions - kept equal without shifting anything. [5], [6]

A. REFERENCE FIR AND DA BASED FIR ANALYSIS

A look at Fig. 3 reveals how a FIR filter behaves when multipliers are part of its setup - this version becomes our baseline. Despite differences in design, the structure here sets the standard others will be measured against.[8]

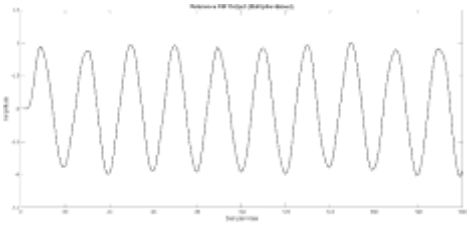

Fig. 3:Reference FIR Output(Multiplier Based)

Fig. 4 shows how the result lines up against the DA-driven FIR filter.

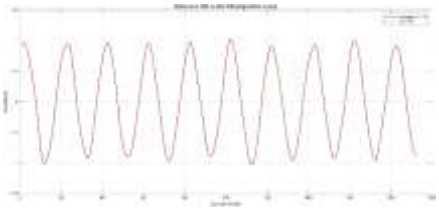

Fig . 4 : Reference FIR vs DA FIR

Looking at the graph, it becomes clear the two outputs match exactly. Few realize the outcome matches a standard FIR filter, yet multiplier-free design makes it happen through DA methods instead.

Look again at Figure 5 - there it shows what we expected. The outcome matches, meaning our setup works as intended.

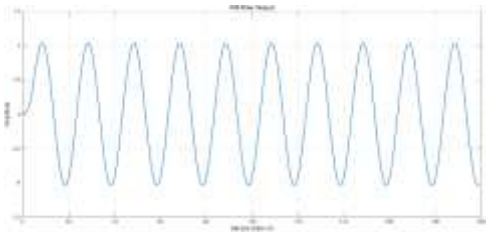

Fig . 5 : FIR Filter Output

B. Adaptive Filtering Using LMS

A twist came when we picked the LMS method to let the filter adjust itself. The way it learns on the fly shapes how cleanly it responds over time.[2], [9]

Fig. 6 gives a look at the LMS filter's result beside the target signal.

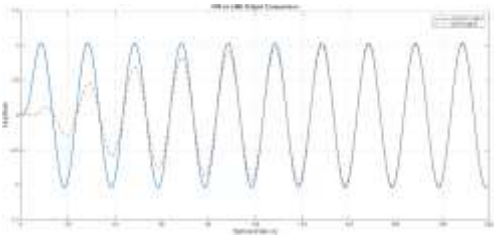

Fig . 6: FIR vs LMS Output Comparison

Every step forward pulls the LMS output nearer to the target signal, though early on it misses the mark. Over time, through steady adjustment, its shape tightens into alignment. This expectation shapes how we see the algorithm.[2] What matters most shows up in its results. Its behavior reflects a clear purpose behind design choices. Each step follows a logic meant to deliver consistency. Predictability becomes obvious after repeated tests.[5], [9]

Look at Fig. 7 - it displays the error signal. Over time, the mistake shrinks, a sign the LMS method does what it should. What stands out is how steady the drop feels.

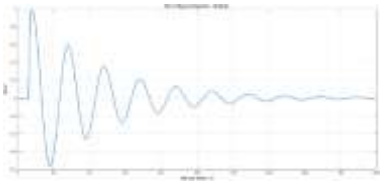

Fig . 7: Error Signal Variation

C. Performance Evaluation(Execution Time and Error)

Execution speed gave us a clue about the filter's performance. The slower it ran, the more we watched for drops in accuracy. MSE crept into view each time results skewed too far from expected values. A tighter runtime often meant cleaner outputs. Errors stood out most when timing lagged behind ideal conditions[8]

Architecture	Execution Time
DA FIR	0.014551
RRPG FIR	0.008106

Table 1: Execution Time Comparison

Looking at Table I, the DA-based FIR filter runs in 0.014551 seconds. In comparison, the RRPG version finishes in just 0.008106 seconds.

Achieving a 44% gain shows the design works better. Efficiency rises because of how it's built.

Accuracy improves when the filter uses the LMS method instead.[1], [10] Down went the mean squared error, dropping from 0.0662 with plain RRPG to just 0.0453 when adding LMS - cutting mistakes by about one third. That shift marks a solid step toward better accuracy, though not every tweak behaves so cleanly under pressure.[3]

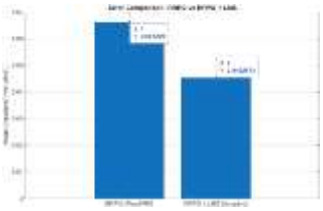

Fig . 8 : MSE Error Comparison

Architecture	Mean Error
RRPG FIR	0.0662
RRPG+LMS	0.0453

Table 2:Mean Error Comparison

It's obvious - the system works better when the adaptive algorithm is included.

D. Hardware Validation

Fairly late into the process came testing, handled through Xilinx Vivado. Hardware got verified on that platform without skipping steps.[3]

Floating around Fig. 9 is what the simulation wave looks like.

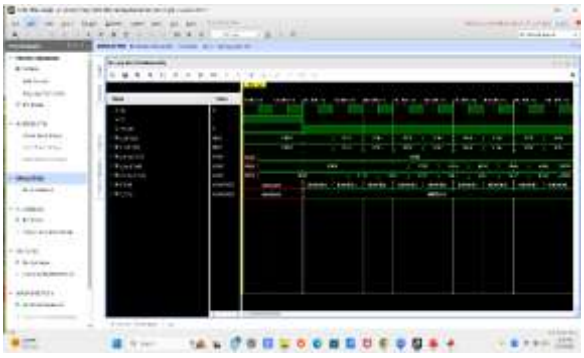

Fig . 9 : Vivado Simulation Waveform

Fig. 10 gives the comparison - hardware versus MATLAB - to verify accuracy. Correctness comes through matching results side by side there.

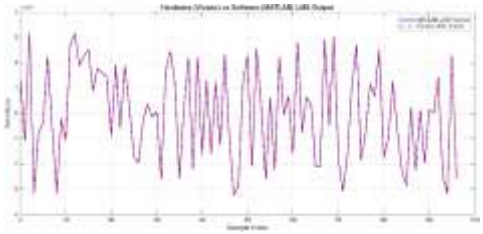

Fig . 10: MATLAB vs Vivado Output

Almost identical results show up here. Still, one might spot tiny differences if looking hard enough.

Fig. 11 reveals how they differ. Nearly no mistake appears.

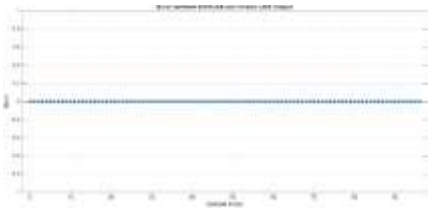

Fig . 11 : Error Between MATLAB and Vivado

Finding lines up with expectations - hardware runs just like the simulation. It behaves as predicted, no gaps showing through.

5. Conclusion and Future Scope

The new RRPG-based adaptive FIR filter is really good because it gets rid of multipliers. This means it works efficiently. The filter also keeps giving results because it uses LMS-based adaptation.[4] When we tested it we found that it works faster and produces fewer mistakes than the old designs. We also checked the hardware. It works the same as the simulation. In the future we can use the RRP-based FIR filter for more complex filters and we can make it use less power. We can also use it in real-time applications such as communication and signal processing systems. The RRP-based adaptive FIR filter will be really useful, in these areas.[4], [11]

REFERENCES

[1] J. A. Jerome, J. G. Daphne, S. Vigneshwaran, and K. Mariammal, "Development and Realization of a Multi-Rate FIR Filter Utilizing Distributed [1] D. Jiménez-Galindo, P. Casaseca-de-la-Higuera, and L. M. San-José-Revuelta, "A Novel Design Method for Digital FIR/IIR Filters Based on the Shuffle Frog-Leaping Algorithm," in *2019 27th European Signal Processing Conference (EUSIPCO)*, Sep. 2019, pp. 1–5. doi: 10.23919/EUSIPCO.2019.8903129.

[2] H. Johnson, "Demonstrating the Potential of Adaptive LMS Filtering on FPGA-Based Qubit Control Platforms for Improved Qubit Readout in 2D and 3D Quantum Processing Units".

[3] G. Singh and N. R. Prakash, "FPGA Implementation of Higher Order FIR Filter," *IJECE*, vol. 7, no. 4, p. 1874, Aug. 2017, doi: 10.11591/ijece.v7i4.pp1874-1881.

[4] M. Arucu and T. Iliev, "Performance Evaluation of FPGA, GPU, and CPU in FIR Filter Implementation for Semiconductor-Based Systems," *JLPEA*, vol. 15, no. 3, p. 40, Jul. 2025, doi: 10.3390/jlpea15030040.

[5] A. Dwivedi, "Enhanced DA Algorithm for FIR Filter Design and FPGA Implementation," *IJRASET*, vol. 12, no. 5, pp. 4483–4488, May 2024, doi: 10.22214/ijraset.2024.62485.

[6] C. Safarian, T. Ogunfunmi, W. J. Kozacky, and B. K. Mohanty, "FPGA implementation of LMS-based FIR adaptive filter for real time digital signal processing applications," in *2015 IEEE International Conference on Digital Signal Processing (DSP)*, Singapore, Singapore: IEEE, Jul. 2015, pp. 1251–1255. doi: 10.1109/ICDSP.2015.7252081.

[7] S. Y. Park and P. K. Meher, "Efficient FPGA and ASIC Realizations of a DA-Based Reconfigurable FIR Digital Filter," *IEEE Trans. Circuits Syst. II*, vol. 61, no. 7, pp. 511–515, Jul. 2014, doi: 10.1109/TCSII.2014.2324418.

[8] G. Gomes, P. Freire, J. E. Prilepsky, and S. K. Turitsyn, "FPGA Implementation of Complex Value-based Clustering Filter for Chromatic Dispersion Compensation in Coherent Metro Links with Ultra-low Power Consumption," Sep. 20, 2024, *arXiv*: arXiv:2409.13381. doi: 10.48550/arXiv.2409.13381.

[9] J. A. A., J. G. D. V., V. S., and M. K., "Development and Realization of a Multi-Rate FIR Filter Utilizing Distributed Arithmetic on FPGA," *JEEA*, vol. 6, no. 4, pp. 343–357, Feb. 2025, doi: 10.36548/jeea.2024.4.006.

[10] D. Jiménez-Galindo, P. Casaseca-de-la-Higuera, and L. M. San-José-Revuelta, "A Novel Design Method for Digital FIR/IIR Filters Based on the Shuffle Frog-Leaping Algorithm," in *2019 27th European Signal Processing Conference (EUSIPCO)*, Sep. 2019, pp. 1–5. doi: 10.23919/EUSIPCO.2019.8903129.

[11] B. Rashidi, F. Mirzaei, B. Rashidi, and M. Pourormazd, "Low Power FPGA Implementation of Digital FIR Filter Based on Low Power Multiplexer Base Shift/Add Multiplier," *IJCTE*, pp. 346–350, 2013, doi: 10.7763/IJCTE.2013.V5.707.