iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** V **Month of publication:** May 2024

DOI: https://doi.org/10.22214/ijraset.2024.62485

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue V May 2024- Available at www.ijraset.com

"Enhanced DA Algorithm for FIR Filter Design and FPGA Implementation"

Akshya Dwivedi[1], Akriti[2], Akash Sharma[3], Aditya Swarnakar[4], Unnati[5]
ABES EC Ghaziabad

Abstract: ***This study looks into the difficulties of using the Distributed Arithmetic (DA) method for Finite Impulse Response (FIR) filters on Field-Programmable Gate Arrays (FPGAs). It focuses on how coefficients are represented, balancing precision using fixed-point arithmetic and quantization. The research explores ways to optimize memory use, aiming to store data more efficiently within FPGA resources and reduce memory needs. To speed up computations, it examines how to make accessing lookup tables faster and suggests improvements in design. The study also considers how to manage FPGA resources effectively, balancing latency, throughput, and resource use. It looks at specific improvements to the DA method for FIR filters to make better use of resources and enhance performance. Overall, this work provides insights and solutions for algorithm design, memory use, lookup table speed, and FPGA architecture to make DA-based FIR filter implementations on FPGAs more efficient.***

I. INTRODUCTION

Digital signal processing (DSP) relies heavily on Finite Impulse Response (FIR) filters for essential tasks. Designing these filters efficiently is crucial, and one

innovative method is using an Improved Differential Evolution Algorithm (DA) to optimize filter coefficients. This combined approach aims to enhance filter performance by reducing passband ripples and stopband distortions. The process starts with simulating the improved DA algorithm in MATLAB or Python to optimize the filter coefficients according to specific criteria. Once optimized, the next step is to implement these coefficients on an FPGA (Field-Programmable Gate Array) using hardware description languages (HDL) like Verilog or VHDL. This step involves creating a hardware architecture that realizes the FIR filter's functionality on the FPGA platform.

The process is iterative and includes testing, validation, and refinement to ensure the filter meets stringent specifications. It combines principles of DSP, optimization techniques, and FPGA design skills to achieve a high-performance FIR filter implementation.

The workflow includes the following key components:

1) *FIR Filter Design using DA:* This involves designing the FIR filter using the Differential Evolution algorithm.
2) *DA Algorithm Optimization:* This stage is dedicated to refining the filter's coefficients using the Differential Evolution algorithm.
3) *FPGA Implementation & Hardware:* This involves implementing the designed FIR filter onto FPGA hardware, including the necessary configurations and connections for execution.
4) *Input/Output Interface Block:* This represents the interfaces for inputting data into the filter and retrieving the processed output signals.

This synergy between DSP and evolutionary algorithms, facilitated by tools like MATLAB or Python, and the subsequent FPGA implementation using HDL, ensures an efficient and high-performing FIR filter.

In recent years, the trend of implementing digital signal processing (DSP) functions, like FIR filters, in Field Programmable Gate Arrays (FPGAs) has grown due to their wide applications in video, audio signal processing, and telecommunications. Traditional FIR filters require multiple multiply-and-accumulate (MAC) blocks, which are resource-intensive and complex to implement in FPGAs. To address this, a multiplier-less architecture called Distributed Arithmetic (DA) has been developed.

DA decomposes MAC operations into a series of lookup table (LUT) accesses and summations, significantly reducing the need for multipliers. However, DA's traditional implementation requires large LUTs, especially for high-order filters, leading to impractical memory sizes. This challenge is mitigated by splitting large LUTs into smaller ones, using offset binary coding, and exploiting symmetrical properties to reduce memory size.

Despite these optimizations, DA still faces challenges as the number of filter taps increases, particularly when the number of taps is a prime number. To overcome these issues, the proposed hardware-efficient DA architecture further reduces LUT sizes and modifies filter structures to enhance speed and reduce resource usage. This new architecture, implemented on a 4VLX40FF668 FPGA device and synthesized with ISE 7.1, demonstrates improved speed and reduced resource consumption compared to previous DA implementations.

A. Objective & Goal

Traditionally, implementing a K-tap FIR filter directly on an FPGA involves using K multiply-and-accumulate (MAC) blocks. These blocks are demanding in terms of resources and logic complexity. To overcome this challenge, a multiplier-free architecture known as distributed arithmetic (DA) has been introduced.

DA substitutes the multiplication operations in the MAC process with a sequence of lookup table (LUT) accesses and additions, employing a bit-serial approach to calculate the inner product of two vectors within a fixed number of cycles. The initial DA architecture stores every possible binary combination of the filter coefficients in a memory or LUT. However, as the number of taps (L) increases, the memory size required grows exponentially, making it impractically large.

To address this, DA employs 2's complement binary representation, allowing pre-computation and efficient storage of data in LUTs. This characteristic makes DA particularly suitable for LUT-based FPGA architectures. However, DA faces a significant issue: the LUT capacity required increases exponentially with the number of filter taps. Specifically, DA implementations need 2^K words of memory, where K is the number of taps in the filter. This requirement becomes even more challenging if K is a prime number, leading to higher hardware resource consumption.

In summary, DA provides an efficient method for implementing FIR filters in FPGAs, but the exponential increase in LUT requirements with higher filter orders remains a major obstacle.

II. LITERATURE SURVEY

In their paper published on August 7, 2017, Sumbal Zahoor, Shahzad Naseem, and Wei Meng (Reviewing Editor) propose a novel approach for designing and implementing efficient FIR digital filters. The paper discusses various design methods for FIR filters using different window functions. It examines and compares bandpass filters of orders 38 and 48, concluding that the Kaiser window yields superior results.[1]

In April 2017, S. R. Reddy and P. Jayakrishnan presented their work on designing FIR filters using an improved distributed arithmetic (DA) approach for high-speed ground-penetrating radar (GPR) systems. Their paper introduces an enhanced DA method to implement high-order digital FIR filters with reduced logical delay and hardware usage. The process begins with the design of a parallel DA, which is then improved through LUT decomposition. The improved DA FIR filters are subsequently implemented on a Xilinx Kintex-7 FPGA chip and used to process radar signals in high-speed GPR systems.[2]

Bo Hong proposes the implementation of FIR filters on FPGA using the DA-OBC algorithm, emphasizing that digital filters are fundamental components in numerous digital signal processing systems. These filters have extensive applications in communication,

$$y[n] = \sum_{k=0}^{L-1} w[k]x[n-k]$$

image processing, and pattern recognition. The hardware implementation of FIR filters utilizes FPGA chips, employing the Kaiser window for design. The unit impulse response, h(n), is calculated using MATLAB software.[3]

In 2005, Qi Yue and colleagues proposed a low-power FIR filter design based on standard cells. Their paper compares three low-power schemes for the multi-hierarchy pipeline design of fixed-point finite impulse response (FIR) digital filters. They adopt an optimal Canonical Signed Digit (CSD) encoding method to minimize the number of adders and subtractors in the design. Additionally, they designed a 16-bit, 16-tap low-pass FIR filter to evaluate the performance of the three different algorithms.[4]

In March 2014, Kumari K. Dhobi, Dr. K. R. Bhatt, and Dr. Y. B. Shukla proposed the FPGA implementation of FIR filters using various algorithms in their retrospective study. They noted that the distributed arithmetic (DA) structure is easy to implement on FPGAs due to pre-calculated results stored in LUTs, simplifying the design process. They highlighted FPGA as an effective solution for realizing digital signal processing filters. Their simulation results indicated that the Kaiser window method is the most effective windowing technique.[5]

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue V May 2024- Available at www.ijraset.com

A. Background

Traditional implementations of the finite impulse response (FIR) filter equation typically rely on L multiply-accumulate (MAC) units. However, integrating multipliers directly using FPGA logic fabric incurs high costs in terms of logic complexity and area utilization, particularly with larger filter sizes. While modern FPGAs offer dedicated DSP blocks to mitigate this issue, challenges persist in reducing area and complexity for very large filters.

An alternative approach is to decompose the MAC operations into a sequence of lookup table (LUT) accesses and summations, known as distributed arithmetic (DA). This strategy offers a more efficient utilization of resources and can be particularly advantageous in FPGA implementations.

III. IMPLEMENTATION OF FIR FILTER USING DISTRIBUTED ARITHMETIC

A. Distributed Arithmetic FIR Filter Architecture

The Distributed Arithmetic (DA) architecture stands out as one of the most renowned methods for implementing Finite Impulse Response (FIR) filters. It addresses the computation of the inner product equation, particularly advantageous when the filter coefficients are known beforehand, as is the case with FIR filters.

An FIR filter of length K is described as:

$$y[n] = \sum_{k=0}^{K-1} h[k]x[n-k] \quad \text{.....(1)}$$

To facilitate analysis, let's introduce a modified input data sequence denoted as x'[k] = x [n - k]. This modification allows us to rewrite equation (1) as follows:

$$y = \sum_{k=0}^{K-1} h[k].x'[k] \quad \text{..... (2)}$$

Next, we represent the input data using B-bit two's complement binary numbers.

$$x'[k] = -2^{B} \cdot x_B[k] + \sum_{b=0}^{B-1} x_b[k] \cdot 2^b \quad \text{.....(3)}$$

Substitution of (3) into (2) yields

$$y = \sum_{k=0}^{K-1} h[k] \cdot (-2^{B} \cdot x_B[k] + \sum_{b=0}^{B-1} x_b[k] \cdot 2^b)$$

$$= -2^{B} \cdot \sum_{k=0}^{K-1} h[k] \cdot x_B[k] + \sum_{b=0}^{B-1} 2^b \cdot \sum_{k=0}^{K-1} h[k] \cdot x_b[k]$$

$$= -2^{B} \cdot f(h[k], x_B[k]) + \sum_{b=0}^{B-1} 2^b \cdot f(h[k], x_b[k]) \quad \text{..... (4)}$$

We have

$$f(h[k], x_b[k]) = \sum_{k=0}^{K-1} h[k] \cdot x_b[k] \quad \text{.....(5)}$$

The implementation of digital filters using this arithmetic is achieved through the utilization of registers, memory resources, and a scaling accumulator.

The original Look-Up Table (LUT)-based Distributed Arithmetic (DA) implementation of a 4-tap FIR filter (where K=4) is depicted in Figure 1. This DA architecture comprises three main units: the shift register unit, the DA-LUT unit, and the adder/shifter unit.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue V May 2024- Available at www.ijraset.com

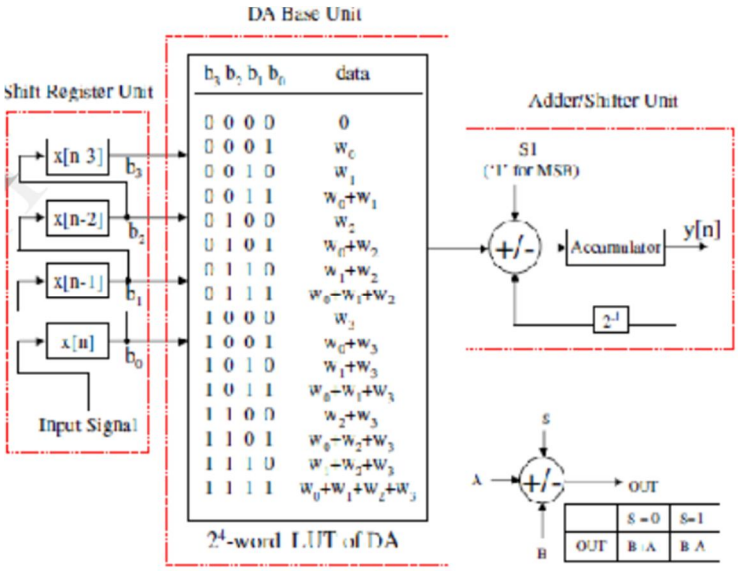


Figure 1. 3 Original LUT-based DA implementation of a 4-tap filter

IV. HARDWARE & SOFTWARE REQUIREMENTS

A. Xilinx Software

Xilinx ISE (Integrated Synthesis Environment) was a software tool developed by Xilinx for the synthesis and analysis of HDL (Hardware Description Language) designs. It was primarily designed for developing embedded firmware for Xilinx FPGA (Field Programmable Gate Array) and CPLD (Complex Programmable Logic Device) product families. However, it's worth noting that Xilinx ISE has been discontinued, with Xilinx Vivado being its successor for FPGA design and synthesis.

B. VHDL

Verilog HDL (Hardware Description Language) serves as a powerful tool for describing digital systems comprehensively. It allows engineers to articulate various digital components such as network switches, microprocessors, memories, and flip-flops effectively. Utilizing Verilog, designers can portray digital hardware systems at diverse levels of abstraction, independent of specific technologies. Designs articulated in Verilog are not only technology-agnostic but also offer ease in design, debugging, and maintenance. Verilog's support for multiple levels of abstraction makes it particularly advantageous for designing large and complex circuits, providing a flexible and efficient platform for digital design and development.

C. Simulation and Results

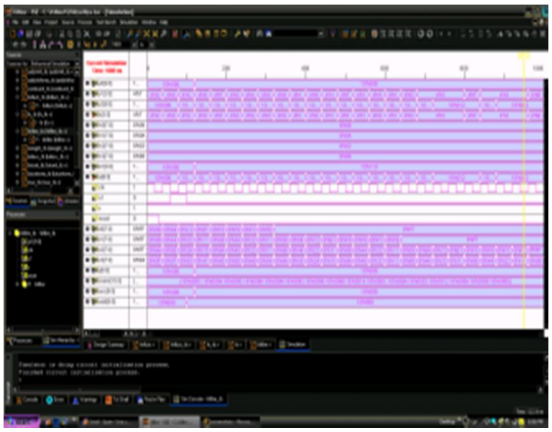


Fig1.1 Original LUT-based DA implementation of a 4-tap filter

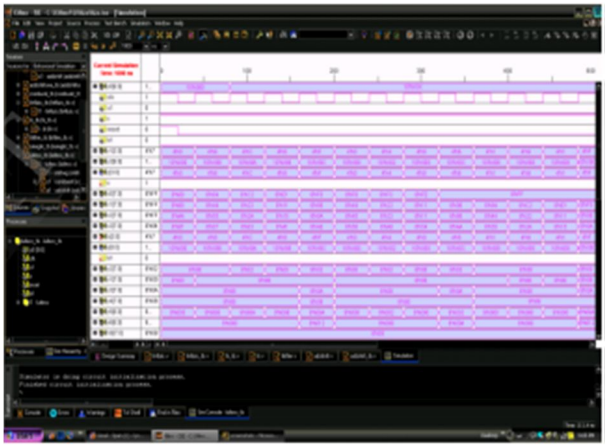


Fig 1.2 LUT-less DA architectures for a 4-tap FIR filter

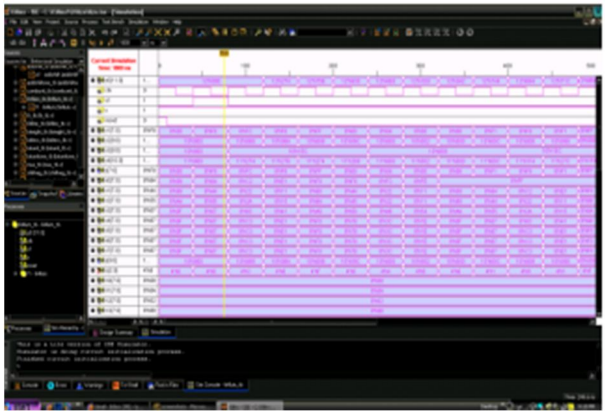


Fig 1.3 4-input look up table architecture for high order filters Without pipelining

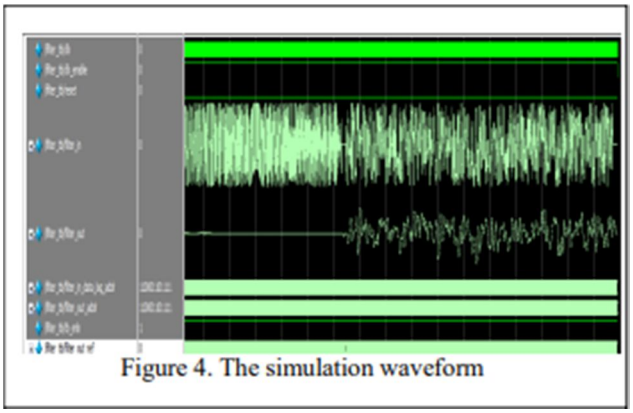


Figure 4. The simulation waveform

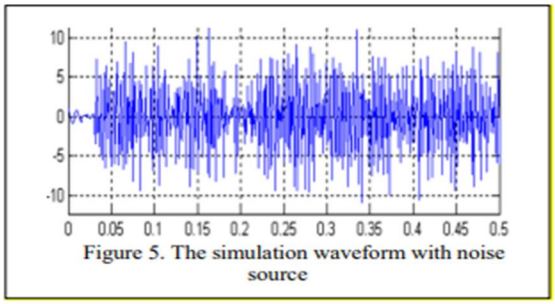


Figure 5. The simulation waveform with noise source

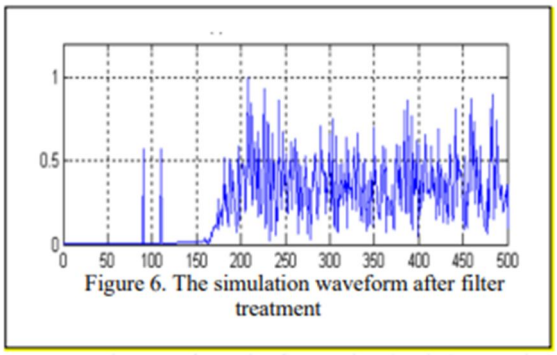


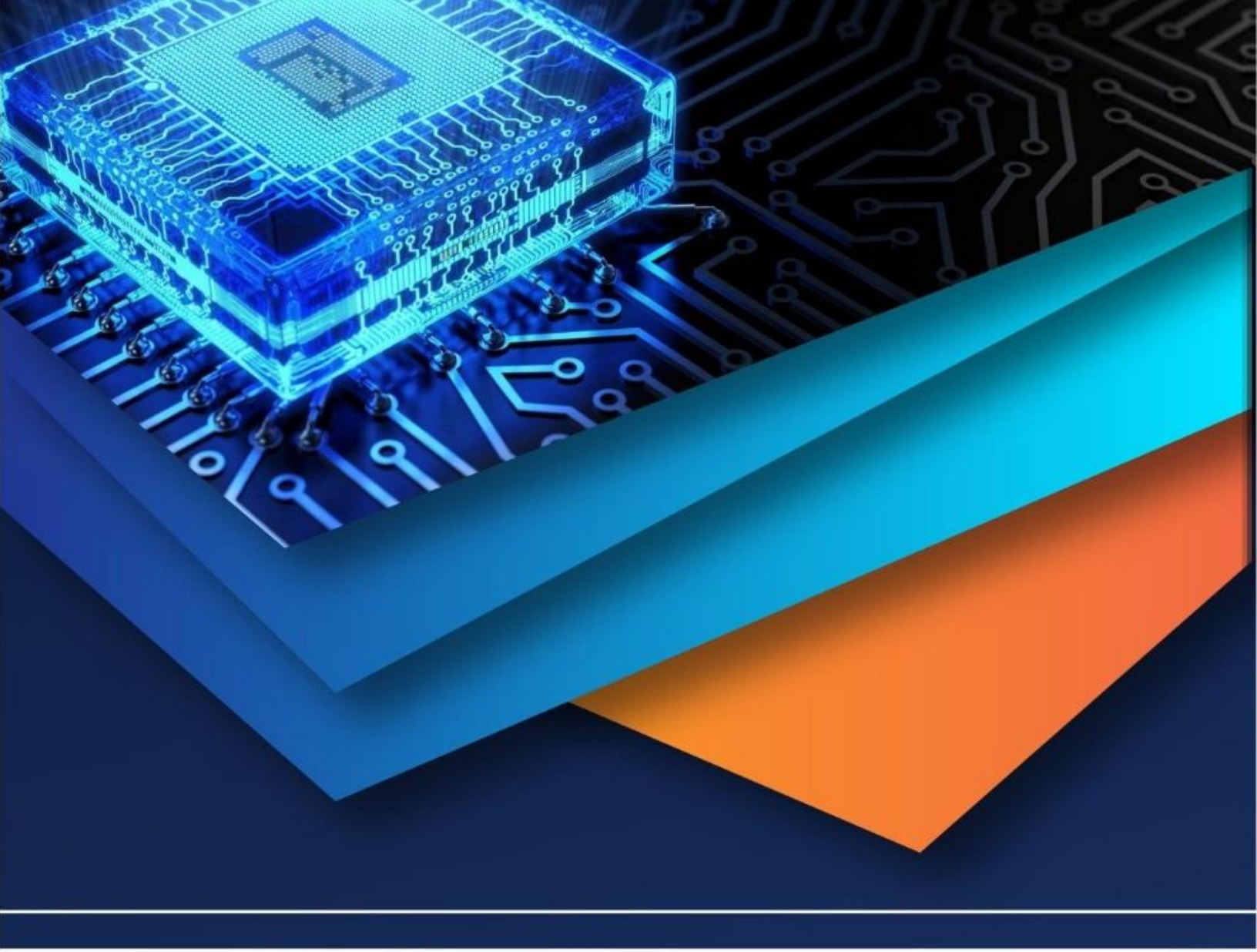
Figure 6. The simulation waveform after filter treatment

V. CONCLUSION

This paper introduces innovative Distributed Arithmetic (DA) architectures tailored for high-order filters. These architectures achieve remarkable reductions in memory usage by halving it at each iteration of LUT reduction, albeit with a minor trade-off in system frequency. Furthermore, the paper proposes partitioning high-order filters into smaller groups, enabling a reduction in LUT size. To achieve high-speed implementation of FIR filters, the paper advocates for adopting a full-parallel version of the DA architecture. Through successful implementation, the paper demonstrates the effectiveness of these DA architectures. Specifically, it showcases the efficient realization of a 70-tap full-parallel DA filter using both an original and a modified DA architecture on a 4VLX40FF668 FPGA device. This implementation underscores the hardware efficiency of the proposed DA architectures for FPGA-based applications.

REFERENCES

[1] Cui W. “Design and Analysis of High Performance FPGA System Timing. Microcomputer Applications”.2014(2)
[2] Fan K.Y et al.”Optimization of FIR filter design scheme based on FPGA”.Research and Exploration in Laboratory.2014,33(5):91- 95.DOI:10.3969/j.issn.1006-7167.2014.05.023.
[3] Li X&Y.T Jiang. “The Digital Filter with Finite Unit Impulse Response Achieved by DSP Window Function Method”. Journal of Science and Economic Market. 2011
[4] Lv W et al. “Optimization Design of FIR Filters Based on FPGA. TV Technology”. 2014,05:71-73.
[5] Qu S.R&J.C P. “A resource optimization algorithm for FIR filter implemented on FPGA.Electronic Design Engineering”. 21(14):147-150.DOI:10.3969/j.issn.1674-6236.2013.14.045.
[6] Wang Z&Q.N.Zhou. “Realization of high order FIR filter based on the improved DA algorithm”.Modern Electronics Technique.2014(4):8-12.DOI:10.3969/j.issn.1004-373X. 04.003.
[7] Yang W.M et al. “FPGA implementation of the distributed structure FIR filter”.Electronic teaching journal.
[8] Zhu X.X, j.Cai&W.Lu. “The design of the filter based on the optimization DA algorithm and its FPGA implementation”. Application of Electronic Technique.41(2):59-60,64.DOI: 10.16157/j.issn.0258-7998.2015.02.012
[9] Cui L&Z.X Zhang. “Realization and comparison of FIR filter based on FPGA design”. Electronic Design Engineering”.20(20):168-170.DOI:10.3969/j.issn.1674- 6236.2012.20.060.
[10] Yong Du. “The implementation of Digital filter based on Matlab and FPGA (Altrra/Verilog)”. Beijing: Electronic Industry Press, 2015.03



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY