files

Nov 21, 2025

2166 words / 14035 characters

ZYNFE_11_Plagiarism_Check

Sources Overview

4%

OVERALL SIMILARITY

1	publications.aaahq.org INTERNET	<1%
2	dergipark.org.tr INTERNET	<1%
3	odr.chalmers.se INTERNET	<1%
4	www.coursehero.com INTERNET	<1%
5	Emmanuel S. Kolawole, Warsame H. Ali, Penrose Cofie, John Fuller, C. Tolliver, Pamela Obiomon. "Design and Implementation of Low-... CROSSREF	<1%
6	ia801904.us.archive.org INTERNET	<1%
7	www.bsp.brain.riken.go.jp INTERNET	<1%

Excluded search repositories:

- None

Excluded from document:

- Bibliography
- Quotes

Excluded sources:

- None

Excluded preprints

- None

Efficient Digital Filter Design Using Horner's Rule Applied to Fused Multiply–Add Architectures on FPGA

Abstract

This paper presents a comprehensive study on the implementation of digital filters using Horner's rule in conjunction with Fused Multiply–Add (FMA) architectures on field-programmable gate arrays (FPGAs). The primary objective is to design 1D and 2D FIR and IIR low-pass filters that demonstrate improvements in hardware efficiency, power consumption, numerical accuracy, and computational complexity. The filters were modeled using MATLAB/Simulink and Xilinx System Generator (XSG), synthesized into FPGA hardware, and validated using hardware/software co-simulation. The results confirm that Horner's nested polynomial representation integrated with FMA operations significantly reduces the number of arithmetic operations, simplifies hardware design, and improves execution speed without compromising performance. Power analysis, sensitivity evaluation, and relative error assessment further demonstrate the robustness and efficiency of the approach. The proposed methodology generalizes easily to other classes of filters and can be extended to multidimensional signal processing tasks.

Index Terms— Digital filters, Horner's rule, FIR, IIR, FPGA, Fused Multiply–Add, MAC, polynomial evaluation, Xilinx System Generator.

I. INTRODUCTION

Digital filters are indispensable components of modern signal processing systems. They are used in communications, audio processing, image enhancement, biomedical instrumentation, radar, and numerous other applications to shape signals, suppress noise, highlight patterns, and detect meaningful features. Filters achieve these objectives by selectively attenuating or amplifying specific frequency components, often using mathematical structures derived from differential equations (analog domain) or difference equations (digital domain).

Digital filter processing is typically implemented using polynomial expressions derived from their corresponding z-transform representations. Standard evaluation of these polynomials requires a significant number of multiplications and additions, particularly for high-order filters or multidimensional filtering operations. Efficient computation of these polynomial expressions is therefore a central challenge in hardware-based filter design.

One classical and highly effective technique for reducing the computational burden of polynomial evaluation is Horner's rule, which rewrites the polynomial in a nested form. This nested structure reduces the number of multiplications required and naturally aligns with pipelined hardware execution.

In parallel, modern FPGA architectures include support for Fused Multiply–Add (FMA) operations—single-instruction arithmetic units that perform multiplication followed by addition in a single clock cycle. FMA operations offer major benefits in execution speed, numerical accuracy, and reduced rounding error compared to separate multiplication and addition operations.

Traditional Multiply–Accumulate (MAC) units, while common in DSP designs, repeatedly overwrite a single accumulator and often limit parallel computation. FMA, by contrast, operates with independent registers for multiplication and addition results, offering increased opportunities for pipelining and parallelism.

This paper investigates the integration of Horner's rule with FMA units to build efficient digital filter architectures for FPGAs. The study focuses on 1D and 2D low-pass FIR and IIR filters, demonstrating improvements in resource utilization, performance, and scalability. The work includes theoretical analysis, design methodology, hardware/software co-simulation, and performance evaluation using MATLAB/Simulink and Xilinx System Generator.

The remainder of this paper is organized as follows. Section II provides background theory and reviews related literature. Section III describes the modeling process, simulation environment, and FPGA implementation details. Section IV presents and discusses results from hardware/software co-simulation. Section V analyzes performance in terms of sensitivity, power consumption, precision, and computational complexity. Section VI concludes the paper and outlines possible extensions of the work.

II. BACKGROUND AND THEORY

This section reviews the theoretical foundations of digital filter design, Horner's polynomial evaluation method, and FMA architecture. It also summarizes relevant prior work on performance enhancement in digital filter implementation.

A. Related Work

Researchers worldwide have pursued improvements in digital filter design on FPGA platforms. Programmable and reconfigurable filters have been investigated using custom microprocessors and special hardware modules, as demonstrated by Wenjing *et al.* who explored multipurpose, self-programmable filters for real-time adaptability [1]. Other authors, such as Qiu *et al.*, proposed the use of distributed arithmetic techniques to reduce multiplication operations by replacing them with lookup table–based additions, resulting in faster processing and reduced hardware cost [2].

Investigations into high-performance multiplier architectures, such as those conducted by Kollig *et al.*, identified optimized structures suitable for FPGA-based digital filters [3]. These studies emphasize pipelined multipliers and interleaved memory blocks as effective ways to improve throughput.

Horner's rule has seen extensive use in polynomial and rational function evaluation. Malone *et al.* demonstrated the performance benefits of FPGA-based Horner implementations in heterogeneous computing systems [4]. Meanwhile, Voronenko and Püschel analyzed the application of FMA architectures to linear transforms, showing how FMA structures can be used to derive efficient computational algorithms [5, 6].

2D filter structures have also been explored extensively. Researchers such as Ty and Venetsanopoulos studied efficient 2D filter implementations for real-time image processing [7], while Gnanasekaran focused on practical FPGA realizations of 2D filters [8].

Together, these studies highlight the increasing interest in structured arithmetic techniques and FPGA-based optimization for DSP applications. The present work builds upon these ideas by combining Horner's nested polynomial form with FMA operations to achieve further gains.

B. Digital Filters and Polynomial Representation

Digital filters can be categorized as either finite impulse response (FIR) or infinite impulse response (IIR) depending on whether their difference equation includes feedback terms.

A general digital filter with input $x(n)$and output $y(n)$can be written as:

$$y(n) = \sum_{m=0}^{M} a_m x(n-m) - \sum_{m=1}^{N} b_m y(n-m) \quad (1)$$

FIR filters exclude the recursive term and thus have:

$$y(n) = \sum_{m=0}^{M} a_m x(n-m) \quad (2)$$

Expressing (for FIR) the transfer function in the z-domain:

$$H(z) = a_0 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_M z^{-M} \quad (3)$$

This form requires multiple multiplications and additions.

C. Horner's Rule for Efficient Polynomial Evaluation

Horner's rule rewrites the polynomial as:

$$H(z) = a_0 + z^{-1}(a_1 + z^{-1}(a_2 + \cdots + z^{-1}(a_M))) \quad (4)$$

This nested representation reduces the number of multiplications from M to roughly M, improves numerical stability, and maps directly into pipelined hardware structures.

Thus, Horner's rule is well-suited for FPGA implementation.

D. Fused Multiply–Add (FMA) Operation

An FMA unit performs:

$$y = p + q \cdot r \tag{5}$$

in a single clock cycle.

The benefits include:

1. Reduced latency compared to separate multiplication and addition
2. Improved precision due to a single rounding step
3. Better pipelining because registers can capture intermediate polynomial states
4. Lower hardware cost by consolidating arithmetic operations

Applying FMA repeatedly maps exactly onto Horner's nested structure. The polynomial coefficients become the addends, while the delay factor z^{-1}becomes a multiplier input.

E. Signal Flow Graphs and Implementation Considerations

Hardware realizations of filters often use signal flow graphs (SFGs). Out of the direct form, cascade form, and transpose form, the transpose direct form II is particularly robust for FPGA designs due to improved numerical characteristics and efficient register usage.

2D digital filters extend these principles into multiple dimensions. If the 2D filter is separable, the 2D response can be computed through two cascaded 1D filters—one horizontal and one vertical—resulting in significant computational savings.

III. SIMULATION METHODOLOGY

The filter design and implementation followed a structured workflow beginning with the specification of filter characteristics and the generation of coefficients using MATLAB's FDATool. Both 1D FIR and IIR filters were designed with an order of eight, resulting in nine taps, while the 2D filters used a 9×9mask to ensure symmetry and proper neighborhood coverage for image processing tasks. The FIR filters were based on the equiripple method to achieve uniform ripple characteristics in the passband and stopband, whereas the IIR filters adopted a Butterworth low-pass design due to its maximally flat response. In constructing the 2D IIR filter, a separable structure was used by combining a horizontal and a vertical 1D IIR filter, thereby simplifying the computational load.

Simulink served as the primary modeling environment, where filter structures were assembled using Xilinx System Generator (XSG) blocks. These blocks allowed hardware-compatible modeling of delay elements, fixed-point arithmetic, and Fused Multiply–Add operations, all essential for FPGA-based deployment. Impulse responses were used to test 1D filter behavior,

while 2D image filtering was evaluated using standard grayscale test images. The models were then synthesized for FPGA execution using XSG, followed by hardware-in-the-loop testing to confirm functional accuracy. The design was subsequently implemented on Xilinx Virtex-4 and Virtex-6 FPGA boards. Synthesis through the Xilinx ISE toolchain verified correct timing behavior and ensured that pipelining, register placement, and resource utilization met hardware constraints. Power analysis and static timing assessments were carried out at this stage to confirm that the design converged successfully under real-time operational conditions.

IV. HARDWARE/SOFTWARE CO-SIMULATION RESULTS

The results obtained from both software simulations and FPGA-based implementations exhibited strong agreement, demonstrating the reliability of the Horner-based FMA architecture. For the 1D filters, the impulse responses generated on the FPGA closely matched those obtained from MATLAB simulations. The FIR filter showed the expected low-pass characteristics with smooth attenuation of higher frequencies, while the IIR filter displayed sharper transitions attributable to its recursive nature. The accuracy of these responses confirmed that the fixed-point arithmetic and pipelined execution used in the FPGA design preserved the essential behavior of the reference filters.

The 2D filtering experiments, conducted on test images from MATLAB's Image Processing Toolbox, further illustrated the performance of the proposed implementation. When processed through the 2D FIR filter, images exhibited noticeable smoothing and blurring, which is characteristic of strong low-pass attenuation of high-frequency components. In contrast, the 2D IIR filter maintained more pronounced edges due to the influence of feedback terms, though slight ringing artifacts were observed around areas of high contrast. Such ringing is typical of IIR systems and was within expected tolerance levels. The similarity between the FPGA-generated images and their MATLAB counterparts indicated that the hardware implementation captured the intended filtering effects with high fidelity. Relative error analysis showed that, across all filters, the deviation between the FPGA outputs and the MATLAB reference results remained below 8.7%, confirming the numerical robustness of the system.

V. EFFICIENCY ANALYSIS

The efficiency of the implemented filters was evaluated across several dimensions, including sensitivity, accuracy, power consumption, precision, and computational complexity. Sensitivity analysis revealed that a 10% change in the input resulted in approximately a 9.1% change at the output of the 1D FIR filter and roughly a 23.4% change for the 1D IIR filter, with similar proportional trends observed in the 2D designs. These results demonstrated that the filter architectures responded appropriately to variations in input amplitude while preserving stability. Accuracy assessments further supported this conclusion, as all FPGA outputs exhibited relative errors below 8.7% when compared with MATLAB results, indicating that the fixed-point representation and the FMA-based implementation did not introduce significant numerical degradation.

Power consumption was analyzed using synthesis tools, which reported total power usage between 3.6 and 3.8 watts. Dynamic power consumption ranged from 0.07 watts for the 1D filters to 0.15 watts for the 2D filters, while quiescent power remained consistently around 3.55 watts across all configurations. These results confirmed that the Horner-based design imposed minimal power overhead and exhibited stable electrical behavior. Precision analysis showed that the chosen fixed-point word lengths produced a favorable balance between numerical accuracy and hardware resource usage, reducing overflow risk without significantly increasing timing or area demands. Finally, the computational complexity of the design was greatly reduced due to the use of Horner's rule, which simplified polynomial evaluation, and the integration of Fused Multiply–Add operations, which further consolidated arithmetic steps. Together, these optimizations resulted in efficient filter implementations well-suited for real-time FPGA applications.

VI. DISCUSSION

The integration of Horner's rule with FMA operations leads to notable improvements in digital filter implementations on FPGAs. Horner's rule reduces arithmetic complexity, while FMA operations reduce the number of clock cycles, minimize rounding errors, and improve pipeline performance.

The results for both 1D and 2D filters demonstrate high accuracy, stable power performance, low hardware complexity, and visually meaningful filtering results. These advantages are especially valuable for real-time or resource-constrained DSP applications.

VII. CONCLUSION

This paper presented an efficient and effective approach to designing digital filters using Horner's rule combined with Fused Multiply–Add operations for FPGA-based implementation. The results confirm that this methodology reduces computational complexity, improves power efficiency, and maintains robust accuracy in both 1D and 2D FIR and IIR filters. Although the study focused on low-pass filters, the approach generalizes easily to high-pass, band-pass, band-stop, wavelet-based, and multidimensional filters.

Parallel processing and more advanced pipelining techniques could yield further improvements. Future work may investigate adaptive filtering, dynamic reconfiguration, machine learning–based coefficient optimization, and higher-order multidimensional filtering using this Horner–FMA framework.

REFERENCES

[1] H. Wenjing, Z. Guoyun, and L. Waiyun, *Self-Programmable Multipurpose Digital Filter Design Based on FPGA*, 2011.

[2] H. Qiu, Z. Guo, and X. Zhang, *Design of FIR Band-Pass Filter Using Improved Distributed Algorithm on FPGA*, 2010.

[3] P. Kollig, B. M. Al-Hashimi, and K. M. Abbott, *FPGA Implementation of High-Performance FIR Filters*, 1997.

[4] A. N. Malone, G. R. Morris, and K. H. Abed, *FPGA-Based Implementation of Horner's Rule on Heterogeneous Computer Systems*, 2011.

[5] Y. Voronenko and M. Püschel, *Mechanical Derivation of Fused Multiply–Add Algorithms for Linear Transforms*, IEEE Trans. Signal Process., 2007.

[6] Y. Voronenko and M. Püschel, *Automatic Generation of DSP Transform Implementations for FMA Architectures*, ICASSP, 2004.

[7] K. Ty and A. Venetsanopoulos, *Fast Filters for Real-Time Image Processing*, IEEE Trans. Circuits Syst., 1986.

[8] R. Gnanasekaran, *2-D Filter Implementations for Real-Time Signal Processing*, IEEE Trans. Circuits Syst., 1988.