



International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.699

Volume 14, Issue 4, April 2025

Optimized FIR Filter Architecture with High Speed Multiplier and Three Operand Adder

Dr.Shaik Khader Zelani

Professor, Department of ECE, Sri Vasavi Institute of Engineering & Technology (Autonomous), Nandamuru, Pedana,
A.P, India

P.Mahathi, B.V.Manikanta, B.J.A.N.V.Narasimha Rao, M.Thiru Venkat, K.N.A.S.Kumar

Department of ECE, Sri Vasavi Institute of Engineering & Technology (Autonomous), Nandamuru, Pedana, A.P, India

ABSTRACT: Finite Impulse Response (FIR) filters find extensive use in digital signal processing (DSP) applications based on their natural stability and linear phase characteristics. The performance of an FIR filter greatly relies upon the efficiency of multipliers and adders implemented within it. This Paper reports an optimized FIR design with an implementation of a high-speed multiplier and new three-operand adder architecture. The Proposed system provides increased computational performance by minimizing latency and power dissipation. Verilog HDL simulations confirm the suitability of the design for real-time signal processing applications. Improvements in area, power, and speed performance compared to traditional FIR filters are reflected in the results.

KEYWORDS: Digital Signal Processing (DSP), FIR Filter, High-Speed Multiplier, Verilog HDL, Three Operand Adder, VLSI, Low Power Design.

I. INTRODUCTION

Finite Impulse Response (FIR) filters are extensively applied in Digital Signal Processing (DSP) applications because they possess linear phase characteristics, inherent stability, and simplicity of implementation. The performance of FIR filter design relies to a large extent on the efficient multiplication and accumulation of filter coefficients. FIR filters are distinct from Infinite Impulse Response (IIR) filters because their response to a unit sample is of limited duration, but IIR filters respond infinitely through internal feedback. FIR filters are non-recursive with no loop of feedback, while IIR filters are recursive and contain feedback. The impulse response of a FIR filter settles to zero eventually, but an IIR filter's impulse response can remain ongoing. Whereas IIR filters are better suited for emulating traditional analog responses, FIR filters are less susceptible to instability, less difficult to control, and are composed of only zeroes compared to IIR filters composed of poles and zeroes. In contemporary VLSI systems, units like multipliers contribute significantly to the performance of FIR filters. High-performance multipliers are required in minimizing computational latency, power, and chip area. To tackle these challenges, approximate multipliers have been proposed for error-resilient use cases where exact precision is not essential-much like multimedia, image processing, and convolutional neural networks (CNNs). These multipliers enhance speed and energy efficiency by reducing logic complexity using methods such as karnaugh Map (K-map) based optimization or Booth encoding. Approximate multipliers generally include three stages: partial product generation, partial product array compression, and final addition. Certain methods incorporate approximation at one or more of these steps using logic simplification and approximate compressors to simplify hardware. Approximate compressors such as 5:3 or 4:2 are employed to compress several bits of partial product row into fewer rows, speeding up the computation. While these approximations save delay and area, they tend to introduce controlled errors, resulting in a trade-off between electrical performance and computational correctness. The utilization of approximate compressors, such as the logic-level simplification-based and error-compensation-based ones, has yielded encouraging results in negotiating this trade-off. Since FIR filters involve extensive use of multiplication and addition operations, the use of optimized arithmetic units can lead to substantial performance improvement. In this work, such integration is investigated to design faster, area-conscious, and low-power FIR filter architectures to be implemented in state-of-the-art embedded systems.

II. PROPOSED SYSTEM

Finite Impulse Response (FIR) filters find extensive applications in Digital Signal Processing (DSP) applications, in which the filter coefficient multiplication and accumulation are a major factor of filter design accuracy. FIR filters are digital filters whose response to a unit sample function is of finite duration, as opposed to Infinite Impulse Response (IIR) filters that respond for an infinite time to a unit impulse. FIR filters are usually implemented with non-recursive methods, i.e., there is no feedback in the system, while IIR filters are recursive and have feedback loops. The impulse response of an FIR filter will eventually be zero, whereas that of an IIR filter may go on indefinitely, possibly "ringing" forever. IIR filters tend to be the choice when trying to replicate traditional analog filter responses because of their form, whereas FIR filters are used for their linear phase response, simplicity in control, inherent stability, and reliance on the input signal alone. However, IIR filters rely on input and output signals and both poles and zeroes, which render them less stable and more difficult to control. To maximize the design of the FIR filter in terms of silicon area reduction, a new method replaces conventional adders with a new three-operand adder in a 9-tap FIR filter structure. The new adder from a parallel prefix-based architecture presents a four-stage logic structure that includes bit-addition logic, base logic, propagate and generate (PG) logic, and sum logic. In the bit-addition phase, full adders carry out bitwise additions of three n -bit inputs to produce sum and carry signals. These are subsequently utilized in the base logic phase to calculate generate (G_i) and propagate (P_i) signals using an array of saltire cells. The base logic also takes into account an external carry input (C_{in}), specifically for determining the first generate signal G_0 . In the third phase, the PG logic calculates the carry bits through black and grey cell logic structures. To consider the performance of this adder, Verilog HDL is utilized to model different three-operand adders such as HHC3A, CS3A, HC3A, and the new design, followed by synthesis with Synopsys Design Compiler under the SAED 32nm CMOS technology. Performance parameters like maximum combinational gate delay, core area, power consumption, Area-Delay Product (ADP), and Power-Delay Product (PDP) are compared. The adder presented in this paper shows considerable performance advantages: it is 3.12 \times , 5.31 \times , and 9.28 \times faster than CS3A for 32-, 64-, and 128-bit inputs, and 1.54 \times , 1.57 \times , and 1.60 \times faster than HC3A respectively. It also decreases core area by 21.2%, 24.1%, and 26.6% over HC3A and decreases power consumption by 25.3%, 26.8%, and 28.2%. Even though it is 12–15% slower than HHC3A, the adder suggested is still better than others in ADP and PDP, and it reduces ADP by 55.1%, 71.6%, and 82.7% compared to CS3A, and 48.9%, 51.7%, and 54.2% compared to HC3A for growing word sizes, and hence is a very efficient option for VLSI implementation in FIR filter structures.

Multipliers are critical building blocks in digital circuits, particularly signal processing, and for designs involving 4:2 compressor-based implementations, partial products are divided into sets of four and compressed column-wise with 4:2 compressors that are generally organized using Wallace tree techniques based on (3,2) counters. The other partial products proceed to the next phase, and even though carry-in and carry-out in 4:2 compressors are not directly observable, they are important factors—either the carry from the input is grounded or managed by employing (3,2) counters at row boundaries. The carry-out of a 4:2 compressor is used as a carry-in for the subsequent stage. Wallace-based implementations further incorporate 5:2 and 6:2 compressors for added optimization. Multipliers can be serial or parallel, and parallel structures such as Wallace tree and Dadda multipliers are much faster than conventional array multipliers because they have logarithmic height with respect to word size. Wallace trees operate on unsigned integers and need a lot of adders in each column to add the partial products. Designers usually implement structural optimizations in order to reduce overall delay, and to achieve this, they employ the Low Area Wallace (LAW) multiplier technique. In this design, the partial products needed for an N -bit multiplier are indicated by dot notation—where every three dots contained within a box is a full adder (FA), and two dots is a half adder (HA), and horizontal lines indicate stages separated in the architecture diagram, such as used by the 8-bit LAW multiplier. Compressors play a central role in this task since they enable multiple partial product bits in the same column to be added at the same time without waiting for the carries from earlier stages. The simplest compressor is the full adder (3:2 compressor), which takes three inputs and produces two outputs (sum and carry). The next one is the 4:2 compressor, which compresses four inputs into two outputs and offers improved compression efficiency. Subsequent improvement is implemented through 5:2 compressors that process five direct inputs (X_1 to X_5) and two carry-in inputs (C_{in1} and C_{in2}), generating two carry-out outputs (C_{out1} and C_{out2}) and a sum and carry output. Historically, 5:2 compressors are synthesized with three cascaded full adders, in which XOR-XNOR logic is employed for the conventional implementation. A ****full adder**** is a digital circuit that sums three inputs—two binary inputs (A and B) and a carry-in (C_{in})—and produces two outputs: the sum (S) and carry-out (C_{out}). Full adders can be cascaded to implement byte-wide or multi-bit adders by carrying the carry from one stage to the next. A ****half adder****, however, is a lesser adder which deals with just two

inputs (A and B) and provides a sum and carry through the use of an elementary XOR gate for the sum and an AND gate for the carry. It cannot deal with a carry-in, but two half adders and an OR gate can be used to generate a full adder. Half adders are constructed from simple logic gates such as XOR and AND, or alternatively from AND, OR, and NOT, which are readily available in integrated circuit (IC) format but will increase the size of the circuit. These minimalistic arithmetic modules—compressors, full adders, and half adders—are collectively the components that make up the efficient multipliers in high-speed digital systems.

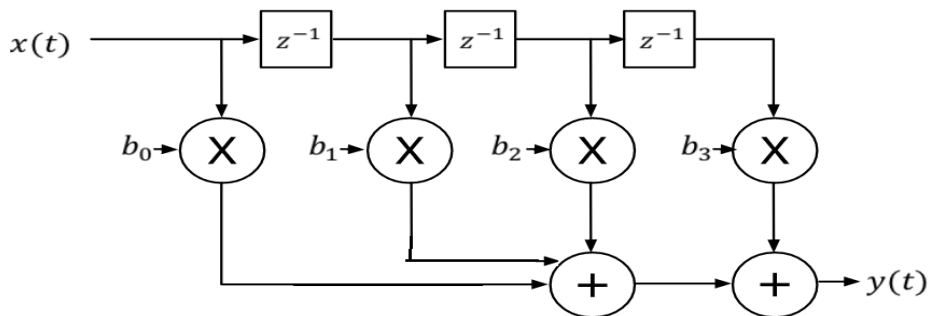


Fig.1. Block Diagram for Proposed System

III. EXPERIMENTAL RESULTS

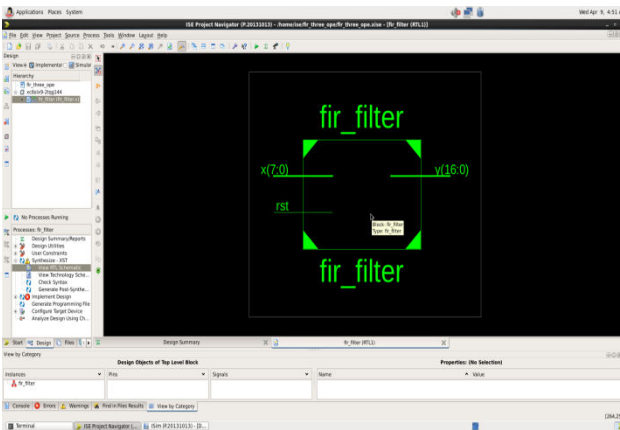


Fig.1. Schematic Diagram of FIR filter

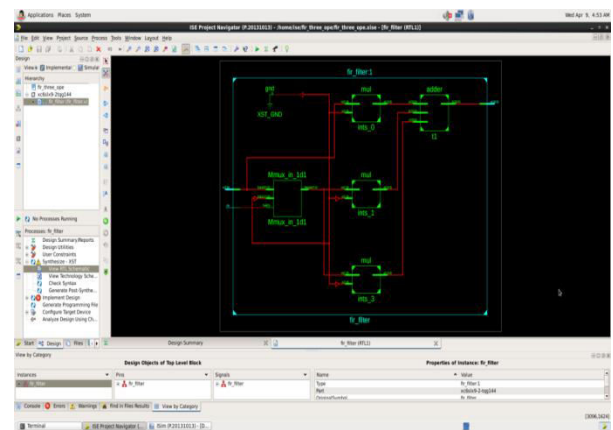


Fig.2. Internal Schematic Diagram of FIR filter

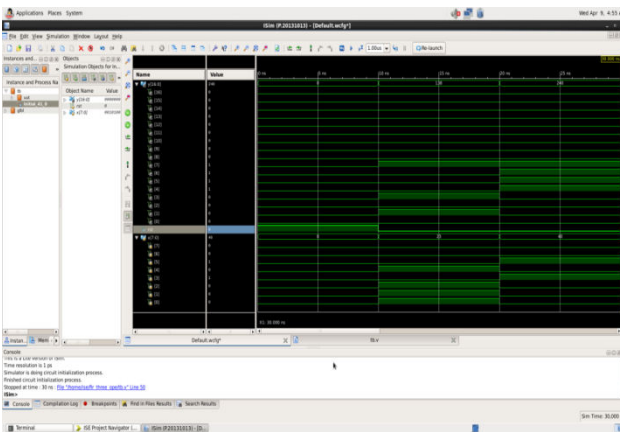


Fig.3. Simulation Results of FIR filter

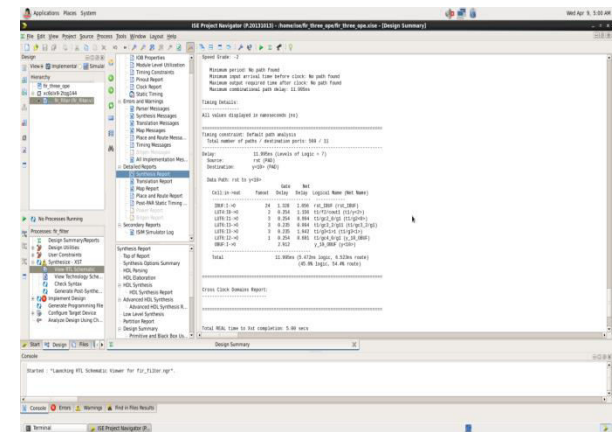


Fig.4. Synthesis Report of FIR filter

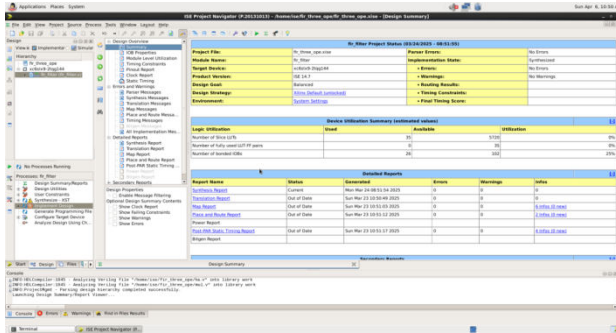


Fig.5. Area Optimization Report

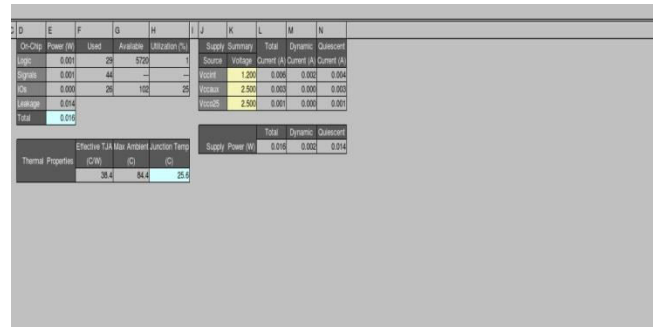


Fig.6. Power Consumption Report

IV. CONCLUSION

We effectively designed and implemented an optimized 9-tap FIR filter structure, utilizing a low-area high-speed Wallace tree multiplier with efficient partial product compression and a new three-operand adder. The new adder, using a four-stage parallel structure, obtained substantial area savings and low power consumption over traditional adders in 32nm CMOS technology. This three operand binary adder is applied for digital signal processing, area efficient system such as satellite, mobile phones. It is appropriate to design the high data rate light weight hardware security system under IoT field. It is developed to minimize area and power by minimizing the number of adder unit. On the basis of this implementation of the FIR filter the Area Delay Product and Power Delay Product are minimized. The path delay is obtained as 5.684n s compared to the others adders it is very smaller. So that it consumes less area and power. In future we can implement in hardware security system. This optimized architecture presents a solution to improve the performance of FIR filters in many applications such as biomedical signal processing and wireless communications.

REFERENCES

- [1] Z. Liu, J. GroBschadl, Z. Hu, K. Jarvinen, H. Wang, and I. Verbauwhede, "Elliptic curve cryptography with efficiently computable endomorphisms and its hardware implementations for the Internet of Things," *IEEE Trans. Comput.*, vol. 66, no. 5, pp. 773–785, May 2017.
- [2] Z. Liu, D. Liu, and X. Zou, "An efficient and flexible hardware implementation of the dual-field elliptic curve cryptographic processor," *IEEE Trans. Ind. Electron.*, vol. 64, no. 3, pp. 2353–2362, Mar. 2017.
- [3] S.-R. Kuang, K.-Y. Wu, and R.-Y. Lu, "Low-cost highperformance VLSI architecture for montgomery modular multiplication," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 24, no. 2, pp. 434–443, Feb. 2016.
- [4] S. S. Erdem, T. Yanik, and A. Celebi, "A general digitserial architecture for montgomery modular multiplication," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 5, pp. 1658–1668, May 2017.
- [5] A. K. Panda and K. C. Ray, "Modified dual-CLCG method and its VLSI architecture for pseudorandom bit generation," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 66, no. 3, pp. 989–1002, Mar. 2019.
- [6] A. Kumar Panda and K. Chandra Ray, "A coupled variable input LCG method and its VLSI architecture for pseudorandom bit generation," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 4, pp. 1011–1019, Apr. 2020.
- [7] A. Rezaei and P. Keshavarzi, "High-throughput modular multiplication and exponentiation algorithms using multibit-scan-multibit-shift technique," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 9, pp. 1710–1719, Sep. 2015.
- [8] A. K. Panda and K. C. Ray, "Design and FPGA prototype of 1024-bit Blum-Blum-Shub PRBG architecture," in *Proc. IEEE Int. Conf. Inf. Commun. Signal Process. (ICICSP)*, Singapore, Sep. 2018, pp. 38–43.
- [9] K. S. Pandey, D. K. B. N. Goel, and H. Shrimali, "An ultra-fast parallel prefix adder," in *Proc. IEEE 26th Symp. Comput. Arithmetic (ARITH)*, Kyoto, Japan, Jun. 2019, pp. 125–134.
- [10] F. Jafarzadehpour, A. S. Molahosseini, A. A. Emrani Zarandi, and L. Sousa, "New energy-efficient hybrid wide operand adder architecture," *IET Circuits, Devices Syst.*, vol. 13, no. 8, pp. 1221–1231, Nov. 2019.



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details