ISSN 2277-2685
IJESR/July-September. 2026/ Vol-16/Issue-3/191-195
Buchi Kishor *et. al.*, /International Journal of Engineering & Science Research
*Full Length Article*

# Efficient FIR Filter Design Employing Vedic Multiplier and Carry Lookahead Adder for DSP Applications

**Buchi Kishor[1], V. Rathaiah[2]**
[1]M. Tech Student: GVR&S College of Engineering and Technology, Budampadu-522017, Guntur (Dt), A.P., India.
[2]Assistant professor: GVR&S College of Engineering and Technology, Budampadu-522017, Guntur (Dt), A.P., India.
Mail Id: buchikishor40856@gmail.com[1], vasrathaiah319@gmail.com[2]
***Accepted 02-07-2026***


***Abstract-****The rising requirement for high-performance, low-power digital signal processing (DSP) systems requires efficient arithmetic solutions, notably in Finite Impulse Response (FIR) filters, where multipliers and adders account for most hardware expenses. To boost speed, the current FIR design makes use of Weinberger adders, composite adders, and Weinberger-based multipliers; nonetheless, the intricate carry-generation structure and multi-stage adder trees result in higher LUT utilization and increased dynamic power. This paper suggests an optimized FIR design that combines a high-speed Carry Lookahead Adder (CLA) with a Vedic (Urdhva–Tiryagbhyam) 8×8 multiplier to get around these restrictions. While the CLA greatly reduces carry propagation latency, the Vedic multiplier offers quick parallel partial-product creation with less logic depth. In comparison to the current Weinberger-based structure, the suggested architecture delivers lower dynamic power, fewer LUTs, and less switching activity. The suggested design improves speed, reduces power consumption, and provides an area-efficient solution appropriate for contemporary low-power DSP applications, according to simulation and synthesis findings in Vivado.*

***Key words****: Weinberger Multiplier, Weinberger Adder, Vedic algorithm, carry look ahead adder, FIR Filter VLSI architecture and Verilog HDL.*

## I. Introduction

The multiplier-adder chain's efficiency has a significant impact on FIR filter performance in traditional hardware implementations. The final filter output is produced by multiplying each incoming input sample by a predetermined coefficient and then adding the resultant products using multi-stage adders. Although this method speeds up propagation, the multi-level generate-propagate logic adds more hardware complexity. Additionally, in order to join partial products, Weinberger multipliers need a number of intermediary adders, which increases wiring overhead and logic use. This increases the number of Look-Up Tables (LUTs), dynamic power consumption, and routing congestion in FPGA implementations, all of which restrict scalability and reduce overall system efficiency. As a result, even while Weinberger-based structures provide faster performance, they are not necessarily appropriate for DSP situations with limited resources or power.Researchers have looked at low-power multipliers, hybrid arithmetic units, and other high-speed approximate adders to get around these restrictions. Among them, Vedic mathematical techniques have attracted a lot of attention, especially the Urdhva-Tiryagbhyam (UT) algorithm. The row-by-row multiplication method employed by conventional and Weinberger multipliers is eliminated by Vedic multipliers, which offer a fully parallel partial-product creation process.

The UT technique computes partial products and intermediate sums concurrently, substantially lowering propagation latency and shortening the critical path. Additionally, the regular and modular structure of the Vedic multiplier leads in decreased hardware complexity, making it extremely appropriate for FPGA and ASIC implementations alike. Alongside multiplier tuning, high-speed adder architectures are vital to further enhance FIR filter performance. While simple adders such as ripple-carry adders promise minimal area utilisation, their linear carry propagation leads to considerable delays.

The Carry Lookahead Adders (CLA), on the other hand, speed addition by computing carry bits in parallel using generate–propagate logic, considerably lowering carry-related latency. CLA structures scale well with bit width and provide a favorable balance between area and speed compared to more

sophisticated prefix adders such as Kogge-Stone or Brent-Kung designs. By replacing the adder chain in the FIR design with a CLA, overall addition time is lowered, giving higher throughput for real-time DSP applications.

## II. Literature Survey

Real-time calculation of processes including filtering, correlation, convolution, and spectrum analysis is crucial to digital signal processing (DSP) systems. Because of its assured stability and adaptability in attaining linear-phase properties, Finite Impulse Response (FIR) filters continue to be fundamental among them. Optimising the hardware implementation of FIR filters to obtain improved performance with lower power consumption and minimal space use has been the subject of extensive research over the past 20 years. Since multipliers and adders are the fundamental components of FIR computing, a great deal of research has been done to evaluate and improve arithmetic unit efficiency, which has a direct impact on digital filters' throughput, latency, and resource usage. This study of the literature examines significant advancements in arithmetic architectures that are pertinent to FIR design, emphasising their benefits, drawbacks, and contributions to high-speed and low-power DSP systems.In DSP, communication systems, biomedical instruments, image processing, audio processing, radar systems, and embedded applications, filters are some of the most essential parts. Their main goal is to keep the necessary information while eliminating undesirable frequency components. From analogue implementations to highly optimised digital hardware solutions utilising FPGA and ASIC technology, filter topologies have changed throughout time. Improving filter performance in terms of speed, power consumption, area utilisation, accuracy, and scalability has been a constant priority for researchers. The main advancements in filter topologies, implementation strategies, optimisation approaches, and current trends in digital filters are reviewed in this overview of the literature.

## III. Fir Filter Using Vedic Multiplier And Carry Look Ahead Adder

The suggested FIR filter executes each multiply-accumulate (MAC) operation using an 8×8 Vedic multiplier succeeded by a carry-lookahead adder for summation. The Vedic multiplier concurrently produces partial products for each input-coefficient combination, which then feed into a CLA tree (or a series of interconnected CLAs) to create the adder-accumulator phase. In scenarios when many partial products need to be aggregated (such as in pipelined MAC stages or across taps), the CLA hierarchy supersedes the composite adder tree used in the baseline. The architecture is modular, including an 8×8 Vedic multiplier unit, parameterizable n-bit carry-lookahead (CLA) modules, pipeline registers, and control circuitry.The Vedic multiplier architecture includes a Carry Look Ahead Adder (CLA) to solve design restrictions. The primary benefit of the Carry Lookahead Adder lies in its simultaneous carry propagation, which significantly lowers power consumption and spatial requirements.

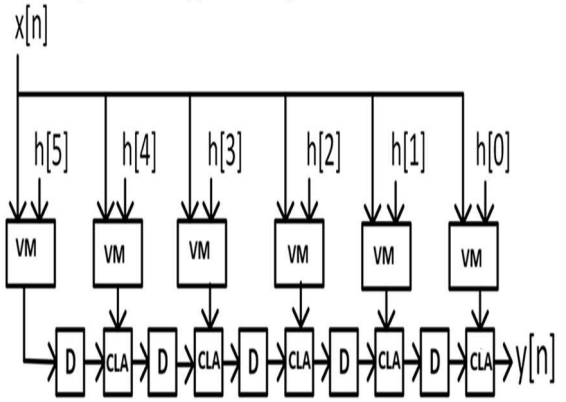


**Fig 1. FIR Filter using Vedic Multiplier and Carry Look ahead Adder**

### A) Modified Vedic Multiplier Design

Steps involved in the Modified Vedic Multiplier Design:

1. Partial Product Generation:
2. Similar to the conventional Vedic multiplier, partial products are produced by multiplying each digit of the two operands. The Vedic multiplication technique breaks this down into smaller, more digestible components, expediting the process.
3. Partial Product Addition using Carry Look ahead Adder.
4. The partial products are summed via a CLA rather than a Ripple Carry Adder. The CLA employs a parallel prefix architecture to concurrently produce carry signals, hence minimising carry propagation duration.
5. The adder utilises a sequence of prefix operations to calculate the total with logarithmic time complexity, making it much more efficient than the Ripple Carry Adder for extensive bit-widths.
6. Final Product Generation:

Upon the completion of the addition phase with the Carry Lookahead Adder, the ultimate result is produced by amalgamating the sums derived from the partial product summation.

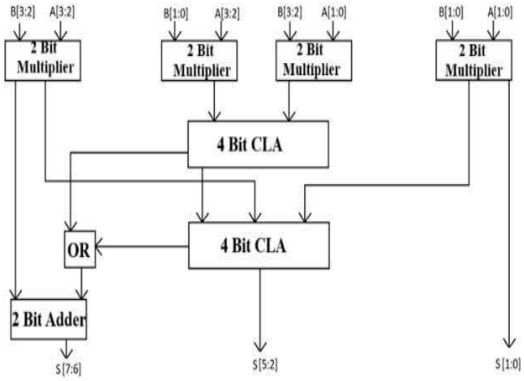


**Fig2. 4 Bit vedic multiplier using CLA**

The method used by the Vedic multiplier is Vedic mathematics. Employing this method will enhance performance while utilizing fewer hardware components. The sutra used by the Vedic multiplier is Urdhva Tiryakbhyam, signifying both vertically and across. The figure illustrates the block diagram of a 4-bit Vedic multiplier circuit. Two comparable parts of the two input bits enable vertical and cross product calculations as seen in the figure using inputs A[3:0] and B[3:0]. The intermediate addition stages employ two adders, as seen in the picture. The output carry Cout from these two adders serves as input to an additional CLA. If bits differ in size, concatenate.A 4-bit Modified Vedic multiplier sends parallel adder outputs into an OR gate, reducing the final Carry Lookahead Adder (CLA) size by 50%.

**B) Carry Lookahead Adder(CLA)**

To minimise calculation duration, more efficient methods for summing two binary values may be used via the utilisation of carry lookahead adders. They operate by generating two signals, P and G, referred to as Carry Propagator and Carry Generator. The carry propagator progresses to the next stage, whereas the carry generator functions to create the output carry, irrespective of the input carry. The diagrammatic illustration of a 4-bit Carry Lookahead Adder is shown below.

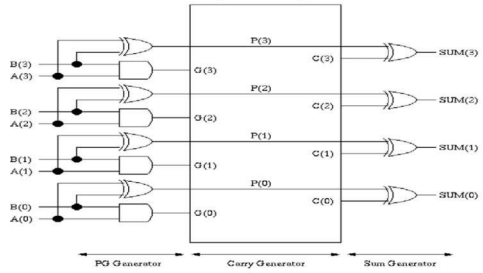


**Fig 3. 4bit Carry Lookahead Adder**

1. Preprocessing: This step involves computation of generate and propagate signals corresponding too each pair of bits in A and B.

$$pi = Ai \; xor \; Bi$$

$$gi = Ai \; and \; Bi$$

2. Carry look ahead network: This element differentiates KSA from other adders and acts as the main catalyst for its exceptional performance. This stage encompasses the computation of carries related to each individual bit. It uses group propagation and generation as intermediate signals.

$$Gi:j = Gi:k+1 \; or \; (Pi:k+1 \; and \; Gk:j \; )$$

3. Post processing: This is the concluding phase and is prevalent across all adders under this category (carry look forward). It entails the calculation of summation bits.

$$Si = pi \text{ xor } Ci\text{-}1$$

## IV. Results

RTL Schematic**:** The RTL schematic, short for register transfer level, is the architectural blueprint and is used to validate the specified architecture against the desired ideal architecture for development. The HDL language facilitates the transformation of an architectural description or summary into a functional summary via the use of coding languages like as Verilog and VHDL. The RTL schematic delineates the internal connection blocks for further analysis. The illustration shown below displays the RTL schematic diagram of the constructed architecture.

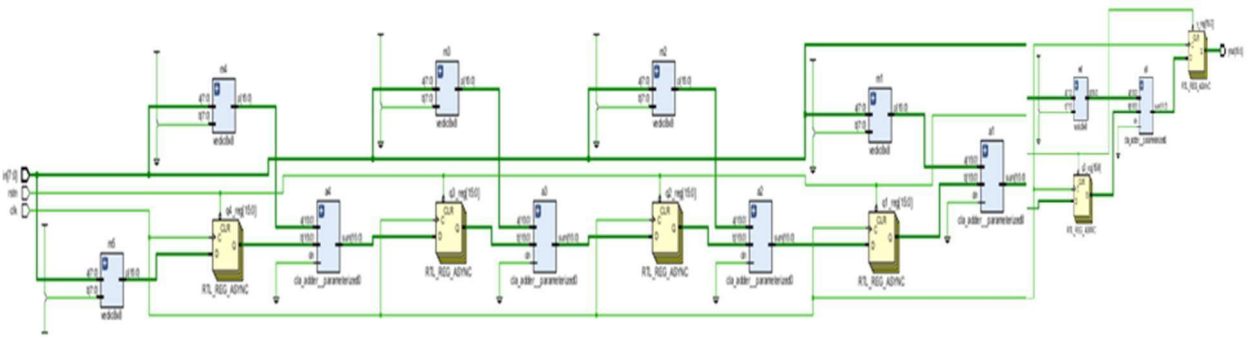

**Fig 4. RTL Schematic of proposed design**

Parameters: In VLSI, the parameters analysed are area and power, which allow for the comparison of one design against another.

**Table I Parameter comparison**

| Parameter | Existing design | Proposed design |
|---|---|---|
| No of LUTS | 93 | 86 |
| Power(mW) | 16.186 | 15.912 |

Simulation:The simulation is the ultimate validation for its functionality, while the schematic serves to verify the interconnections and components. The simulation window is initiated by transitioning from implantation to the simulation on the tool's home page, and it restricts the output to waveforms. This offers the versatility of accommodating many radix number systems.

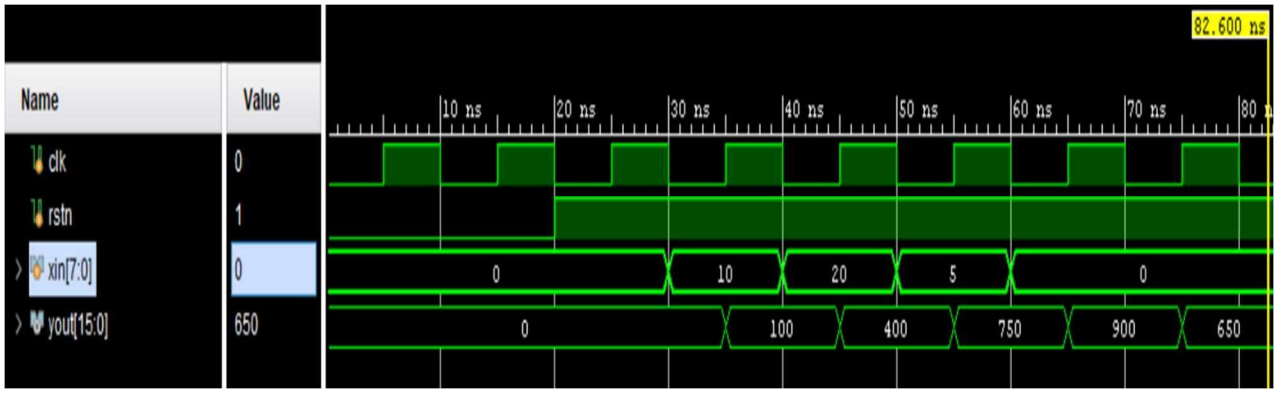


**Fig5. Simulated wave form of proposed design**

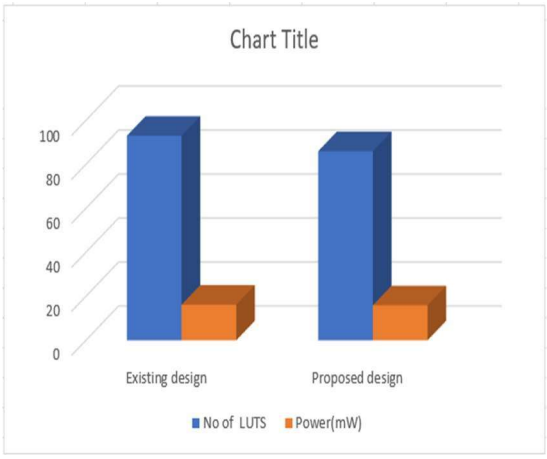


**Fig 6. LUT and Power comparison bar graph**

## Conclusion

The proposed FIR filter architecture, designed using a Vedic Urdhva–Tiryagbhyam multiplier and a Carry Lookahead Adder (CLA), successfully addresses the limitations observed in the existing Weinberger-based design. By reducing logic depth, minimizing partial-product overlap, and optimizing carry propagation, the architecture achieves notable improvements in FPGA implementation metrics. The results demonstrate a 7.5% reduction in LUT utilization, lower dynamic power consumption, and a 56% faster implementation time compared to the baseline system. These enhancements validate the effectiveness of the Vedic–CLA arithmetic combination in delivering a compact, energy-efficient, and high-performance FIR filtering solution. The optimized architecture is well-suited for real-time digital signal processing applications, particularly in power-constrained and resource-limited environments such as biomedical systems, wireless sensor networks, IoT edge devices, audio processing units, and portable communication modules. Its fully LUT-based structure, absence of DSP block usage, and simplified routing make it adaptable to a wide range of FPGA platforms, including low-cost devices used in embedded systems.

In addition to its demonstrated benefits, the architecture opens several avenues for future research. Further optimization can be explored by extending the design to higher-order FIR filters or by applying pipelining to enhance operating frequencies for high-throughput DSP applications. ASIC-level implementation may yield ultra-low-power versions suitable for wearable and implantable medical devices. The architecture can also be adapted for reconfigurable or adaptive FIR filtering, enabling multi-standard communication support. Integration with machine learning accelerators for preprocessing and feature extraction offers another promising direction, especially for edge-AI systems. Moreover, hybrid arithmetic designs combining Vedic multipliers with advanced adders such as Kogge–Stone or Brent– Kung

may further enhance performance.

**References**

**[1]** Arkadiy Morgenshtein, Alexander Fish & Israel A. Wagner, "Gate Diffusion Input (GDI): A Power-Efficient Method for Digital Com binatorial Circuits," IEEE Transactions on Very Large Scale Inte gration (VLSI) Systems, Vol. 10, No. 5, pp. 566-581, 2002.

[2] Arkadiy Morgenshtein, Alexander Fish & Israel A. Wagner, "Gate Diffusion Input (GDI)- A Technique for Low Power Design of Digital Circuits: Analysis and Characterization," IEEE International Symposium on Circuits and Systems, pp. 477-480, 2002.

[3] Khuraijam Nelson Singh & H. Tarunkumar, "A Review on Vari ous Multipliers Design in VLSI," Annual IEEE India Conference (INDICON), pp. 1-4, 2015.

[4] N. Ravi, Y. Subbaiah, Dr. T. Jayachandra Prasad & Dr. T. Subba Rao, "A Novel Low Power, Low Area Array Multiplier Design for DSP Applications," Proceedings of 2011 International Conference on Signal Processing, Communication, Computing and Networking Technologies (ICSCCN 2011), pp. 254-257, 2011.

[5] E. Jagadeeswara Rao, Durgesh Nandan, R.V. Vijaya Krishna & K. Jayaram Kumar, "A Systematic Journal of Multipliers Accuracy and Performance," International Journal of Engineering and Advanced Technology (IJEAT), ISSN: 2249-8958, Vol. 8, Issue-6S, pp. 965 969, 2019.

[6] Suganthi Venkatachalam, Hyuk Jae Lee & Seok-Bum Ko, "Power Efficient Approximate Booth Multiplier," IEEE International Sym posium on Circuits and Systems (ISCAS), 2018.

[7] V. Thamizharasan & Dr. N. Kasthuri, "FPGA implementation of high-performance digital FIR filter design using a hybrid adder and multiplier," International Journal of Electronics, pp. 1-21, 2022.

[8] Ariprasath. S & Dr. C. Santhi, "Transpose Form FIR Filter Design For Fixed and Reconfigurable Coefficients," International Research Journal of Engineering and Technology (IRJET), Vol. 04, Issue 3, pp. 1857-1862, 2017.

[9] Manish B. Trimale & Purushottam G. Chilveri, "FIR Filter Imple mentation on FPGA Using MCM Design Technique," Proceeding of Second International Conference on Circuits, Controls and Com munications, pp. 213-217, 2017.

[10] A. Weinberger & J. L. Smith, "A Logic for High-Speed Addition," National Bureau of Standards Circular 591, Section 1, pp. 3-12, 1958.

[11] R. Suganya & D. Meganathan, "High-Performance VLSI Adders," 3rd International Conference on Signal Processing, Communication and Networking (ICSCN), 2015.