# FPGA-Based Arithmetic Operator Implementation for FIR Filter Design Using FLUT Architecture

**A. Arun[1,*], M. Thangavel[2], V. Kunavathi[3]**

[1]Assistant Professor, Department of ECE, Knowledge Institute of Technology, Salem, India

[2] Professor, Department of ECE, Knowledge Institute of Technology, Salem, India

[3]PG Scholar, Department of ECE, Knowledge Institute of Technology, Salem, India

Emails: aaece@kiot.ac.in; mtece@kiot.ac.in; kunavathi.venmani@gmail.com

**Abstract**

A Hardened adder with carry logic is commonly used in commercial field-programming gate arrays (FPGAs) to enhance arithmetic performance. The rapid expansion of portable multimedia players and communication systems has boosted the demand for high-speed, energy-efficient Digital Signal Processing (DSP) systems. The Finite Impulse Response (FIR) Filter is an essential component when developing an effective digital signal processing system. The use of a digital FIR filters is a key component in DSP. Digital multiplier and adders that are the most crucial arithmetic units used in FIR filters, determining the entire system's performance. As a result, the low-power design of systems has become a primary performance target. This paper also explores the influence of fLUTs and their interactions with toughened arithmetic. FLUTs (Fracturable-LUT) reduce the area by 15%, complementing the latency reductions offered by hardened arithmetic. An FIR filter based on the Carry-Look-ahead adder (CLA) and multipliers was proposed. The tentative results shows that the FIR filter using proposed multiplier method achieves less amount of delay and power reduction compared to conventional method.

**Keywords:** Carry Look-ahead Adder (CLA); Field Programmable Gate Array (FPGA); Fracturable Look-up table (FLUT)

## 1. Introduction

A field programmable logic block connected by a grid that may be changed in the field to interact with other logical blocks to perform different digital operations [5].An ASIC design for a 64th-order adjustable FIR filter architecture with low power and area, based on the enhanced Multiplier. The Basys-3 FPGA board uses the programmable Vedic FIR filtering design (defined in VHDL) for quick prototyping. The delay in propagation is estimated based on the longest path between the input and output. The power consumption evaluation is performed using a clock frequency of 100 kHz [1]. The Carry Look Ahead Adder (CLA) is evaluated based on its arithmetic performance. The purpose of the CLA adder is to reduce the amount of space occupied by the circuit with increased delay while also reducing the delay with increased area consumption. The proposed adder structure achieves optimal performance, which means that the delay is minimized while consuming the same amount of area as in a conventional adder design. Typically, the size of the look-ahead logic is limited to three carriers. AND Gates with 5 to 6 inputs are required for the following two carry signals, making their implementation in CMOS extremely sluggish due to the stacked transistors in the pull-up or pull-down routes. The carry look-ahead block provides the carry calculation, hence the one-bit adder equations for a CLA are the simplified full-adder equation due to carry calculations are no longer needed [3]. Digital filters are obtained in the Digital Signals Processor (DSP), although DSP-based solutions cannot match the high-speed demands of applications due to their sequential structure [2].Use the method of integer linear programming (ILP) for minimizing the number of adders necessary to create a direct/transposed FIR filter complying with a certain frequency parameter [4]. The adder is constructed using a two-level netlist to interleave corrections throughout the design. The Boolean equations associated with every



DOI: https://doi.org/10.54216/IJWAC.090206
Received: February 04, 2025 Revised: February 28, 2025 Accepted: March 30, 2025

netlist are generated and implemented in CMOS technology at 45nm. The proposed BCD adders perform better speed and power than other designs [10].

Fixed-point arithmetic operations suffer from significant data losses, as do single-precision floating-point operations. Most double precision floating- point arithmetic operations use dual rail coding to execute complete detections and require the system to receive acknowledgment upon complete execution, resulting in a worst-case delay regardless of the actual completion time [11]. The approximate recoding adder reduces energy consumption, area, the critical path. An FIR adaptive filter for reducing partial products (PP) and accumulating circuits is designed using distributed arithmetic, signed 32-bit, sixteen-bit radix-8 Booth algorithms, and approximate computing under an insignificant adder. Design and implement a high-productivity, space-effective AES encryption algorithm using an FPGA for safety reason [8].

## 2. Related Work

FIR filters have become common in many fields of processing digital signals due to their ability to provide linear phase as well as system stability. A 70-tap low-frequency FIR filter is employed. The sampling frequency was set to 40 MHz, the bandwidth was 2 MHz, and the precisions of the input and filter coefficients were 13 and 12, respectively. To achieve optimal filtering performance and resource efficiency, use 4-input LUT units with additional multiplexing devices and complete adders. The symmetrical design of the 70-tap FIR filter allows for a reduction of 35 taps, then separated the 35-tap filter into 7 small filters, each with five DA-LUT units. A 4-input LUT combined with a 2x1 multiplexer, and a full adder might be used to build the 5-tap DA-LUT [7]. Reconfigurable FIR filters are implemented and manufactured utilizing 40 nm CMOS semiconductor technology. The suggested design outperforms existing approaches in area-delay products and power performance, which makes it appropriate for low-power filtering applications [13].

The subsequent processing unit renders the spintronic adders immune to input scheduling. The proposed adder requires at least 29% less space. It also provides a 40% shorter carry propagation latency and 37 percent lower power-delay products than the efficient ones [9]. Based on the cache conflict data, an integer linear programming (ILP) approach was proposed to produce the best constant function for system performance. Furthermore, to enhance the scalability of the suggested allocation technique for an enormous task set, providing an interference ratio allows for determining the interference impact quantitatively [12]. The FPGA architecture used in this paper is a heterogeneous architecture with soft logic blocks, simple Os /adjustable memory, fracturable multipliers, and I. A 50% depopulated crossbar connects block inputs and basic logic element (BLE) outputs to BLE inputs, providing internal block communication. We picked a depopulated crossbar because this is usual in most commercial devices. Hardening adders and carry chain dramatically increased the precision of arithmetic operations. Delay reduction was increased by 69%-79%, with an average circuit delay improvement of 13%-16%. While more complicated architectures that harden a CLA enhance standalone adder speed, we discovered that simple hardened ripple-carry adders performed equally well on entire application circuits. We discovered that the greater adaptability of fLUT-based architectures allowed them to be better area-efficient up to 15% [6].

## 3. Proposed Methodology

### A. Baseline Architecture

Fundamental FPGA architecture is built in a 22-nm CMOS processing and is heterogeneous, with soft logical blocks, simple I/Os, customizable memory, and fracturable multipliers [6]. We implemented the carry look-ahead adder to demonstrate the adder structures as shown in figure 1. All 1-bit addition algorithms calculate their output. The look-ahead units are performing their calculations at the same time. Table 1 demonstrates that if a carry occurs in a specific group, it will appear at the left-hand end of that group within at most 5 gate delays and begin propagation throughout the group to its left [3].

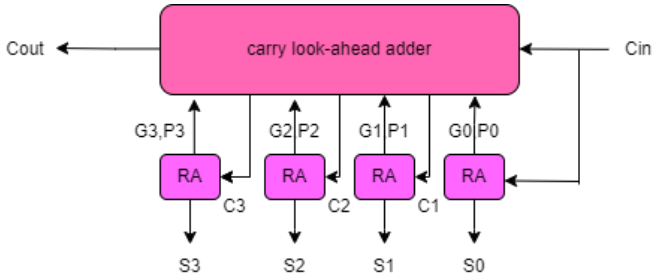


**Figure 1.** Schematic diagram of carry look-ahead adder



DOI: https://doi.org/10.54216/IJWAC.090206
Received: February 04, 2025 Revised: February 28, 2025 Accepted: March 30, 2025

The look-ahead carry logic grows more sophisticated as the number of bits in a group increases. The following equations 1 and 2 compute each positions generate and propagate signals:

$$\text{Generate: } Gi = Gi \bullet Pi \quad (1)$$
$$\text{Propagate: } Pi = Gi + Pi \quad (2)$$

The equations for the carries in a CLA are given by

$$C1 = G0 + P0 \bullet Cin \quad (3)$$
$$C2 = G1 + G0 \bullet P1 + P \bullet \cdot P2 \bullet Cin \quad (4)$$
$$C3 = G2 + G1 \bullet P2 + G0 \bullet P1 \bullet P2 + P0 \bullet P1 \bullet P2 \bullet Cin \quad (5)$$

In equation 3, 4 and 5 attempt to compute the carries further and further in advance, larger gates are required. For example, computing C3 requires the use of a 4 input AND gate and a 4 input OR gate. Hence, usually the size of the look-ahead logic is limited to 3 carries [3]. The CLA employs intermediate information to predict whether a particular bit position will be carried out. Table 1 depicts the truth tables for a full adder, which includes the extra carry information. The bit position cannot be carried out under the delete condition. The propagation condition will only result in a carry out. The generate/propagate condition will always have a carry out at that place.

**Table 1:** Generate and propagate information for a CLA

| A | B | C | Sum | Carry Out | Condition |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | Delete |
| 0 | 0 | 1 | 1 | 0 | Delete |
| 0 | 1 | 0 | 1 | 0 | Propagate |
| 0 | 1 | 1 | 0 | 1 | Propagate |
| 1 | 0 | 0 | 1 | 0 | Propagate |
| 1 | 0 | 1 | 0 | 1 | Propagate |
| 1 | 1 | 0 | 0 | 1 | Generate/propagate |
| 1 | 1 | 1 | 1 | 1 | Generate/Propagate |

### A. Digital Filter Signal Flow Graph

A digital filter modifies signal characteristics in the time and/or frequency domains. A digital signal is created by sampling a continuous signal in multiple periods and then representing the signal as a sequence of discrete values, as opposed to an analog signal, which is continuous and represented as a function of time. The signal flow graph is shown in figure 2.

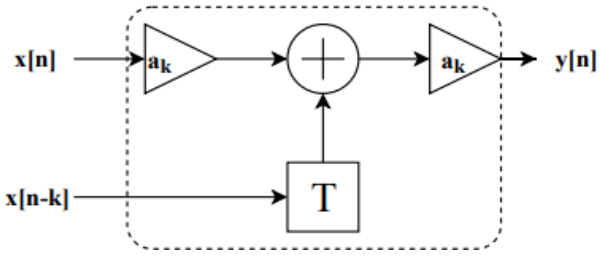


**Figure 2.** Signal flow graph structure



DOI: https://doi.org/10.54216/IJWAC.090206
Received: February 04, 2025 Revised: February 28, 2025 Accepted: March 30, 2025

The difference equation is a mathematical description of digital FIR filter techniques. A linear time invariant (LTI) filter is one of the most often used digital filters. The impulse response of an LTI filter is denoted as h[n]. The impulse response h[n] is the output achieved by using an impulse as an input sequence [14]. The convolution method convolves the impulse response and input sequence in the time domain to produce the output y[n], which may be mathematically expressed as y[n] = x[n] - h[n].

### B. Structure of 4-bit carry Look ahead adder with fLUT

Presenting a method for carrying adaptive structure generation, technology mapping, partitioning, and positioning for any SRAM-based FPGA that can be characterized using the generic models. During each design process, knowledge of the logical structure will be efficiently applied. As previously stated, the duration of a BCLA unit is not limited to a certain value. As a result, we can develop equations for BLOCK-PROPAGATE, GENERATE, and CARRIES.

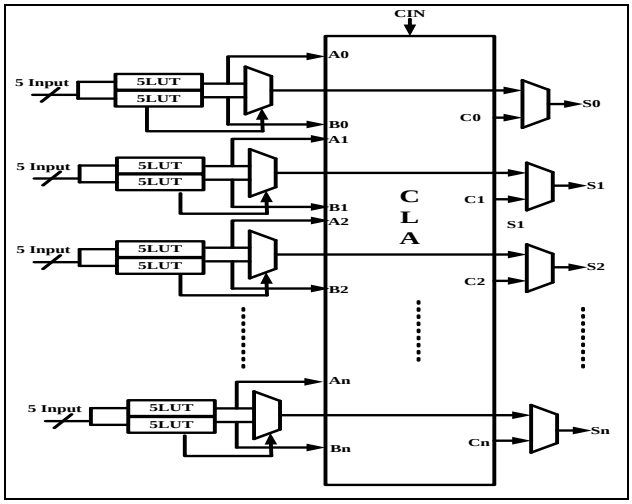


**Figure 3.** Carry look-ahead adder with LUT

The CLA employs additional circuitry to generate carry bits in parallel, hence eliminating the time required to compute the bigger value bit result. Figure 3 depicts the architecture of four-bit CLA. CLA can be divided into two parts: the Partial Full Adder (PFA) and the Carry Look-ahead logic. The PFA generates propagate signals $p_i$, generate signals $g_i$, sum output $s_i$, and carry-out bits $C_i$ +1 via the look ahead logic circuit. In a partial full-adder, propagate P and generate G rely solely on the input bits, while the carry generator is not affected by previous carry-ins. After computing C0, C4 can reach a stable state without waiting for C3 to propagate.

### C. FIR filter using CLA adder design

The FIR filter has three primary components: A D-FF to create a basic delay [15]. A multiplier is a device that is used to carry out the filter coefficients. An added that sums all the nodes at the completion of each tap [16]. Functional verification of all adders and multipliers is carried out is shown in figure 4.

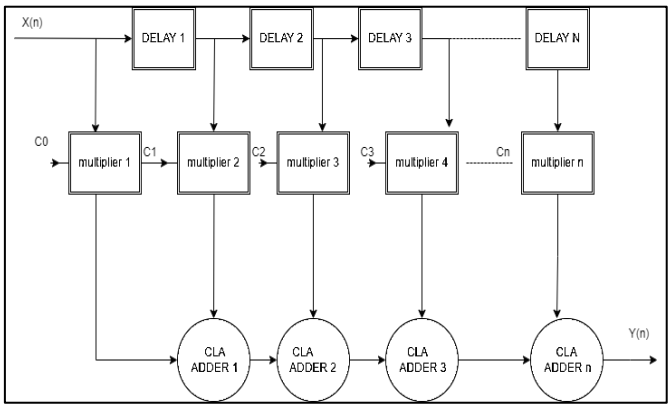


**Figure 4.** Direct method FIR filter with CLA



DOI: https://doi.org/10.54216/IJWAC.090206
Received: February 04, 2025 Revised: February 28, 2025 Accepted: March 30, 2025

In this case, X(n) represents the input filter coefficients, C0, C1, C2, and Cn are the transfer function coefficients, and Y(n) is the output filter coefficient. Multipliers can be replaced by a shift and add operation, known as MCM (Multiple Constant Multiplication), in which a set of constants (here h 0, h 1) is multiplied by a variable (here x(n)).

## 2. Results and Discussion

Concentrate on comparing the various hard adder versions using full application evaluations. These allow us to assess the general quality of the various implementation techniques, considering the total delay and area. Architectures that reduce the area-delay product are the most efficient.

The benchmarks are comprehensive application circuits that can perform a wide range of arithmetic operations such as multiplication, addition, and subtraction using CLA adders with FIR filter. Vivado 2019.1 is used for design, synthesis and implementation.

### A. Schematic Diagram of 4-bit CLA Adder

Figure 5 shows that the representations of 4 bit carry look-ahead adder design.

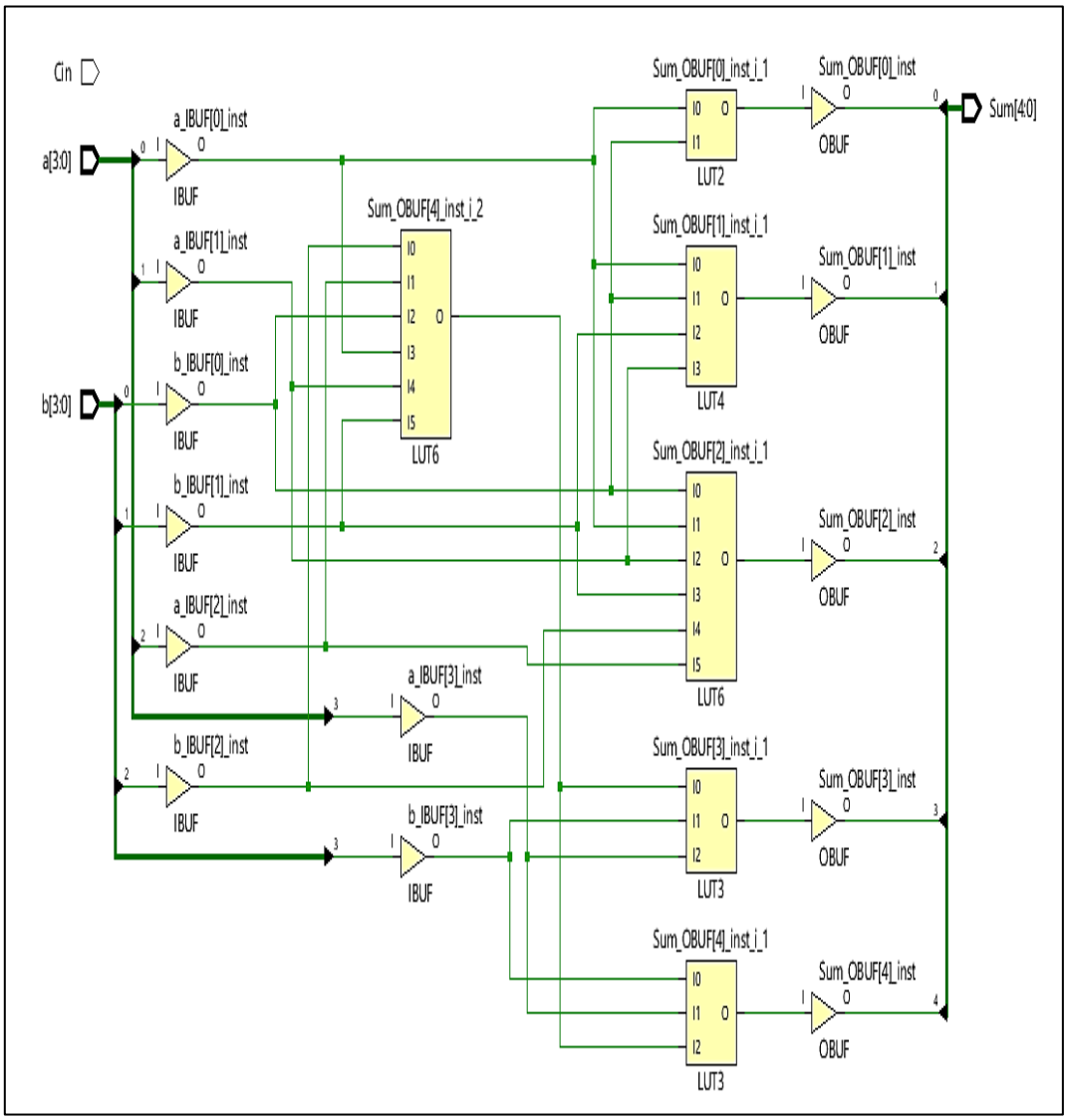


**Figure 5.** Schematic diagram of CLA_4 bit



DOI: https://doi.org/10.54216/IJWAC.090206
Received: February 04, 2025 Revised: February 28, 2025 Accepted: March 30, 2025

### B. Output wave for FIR Filter Design

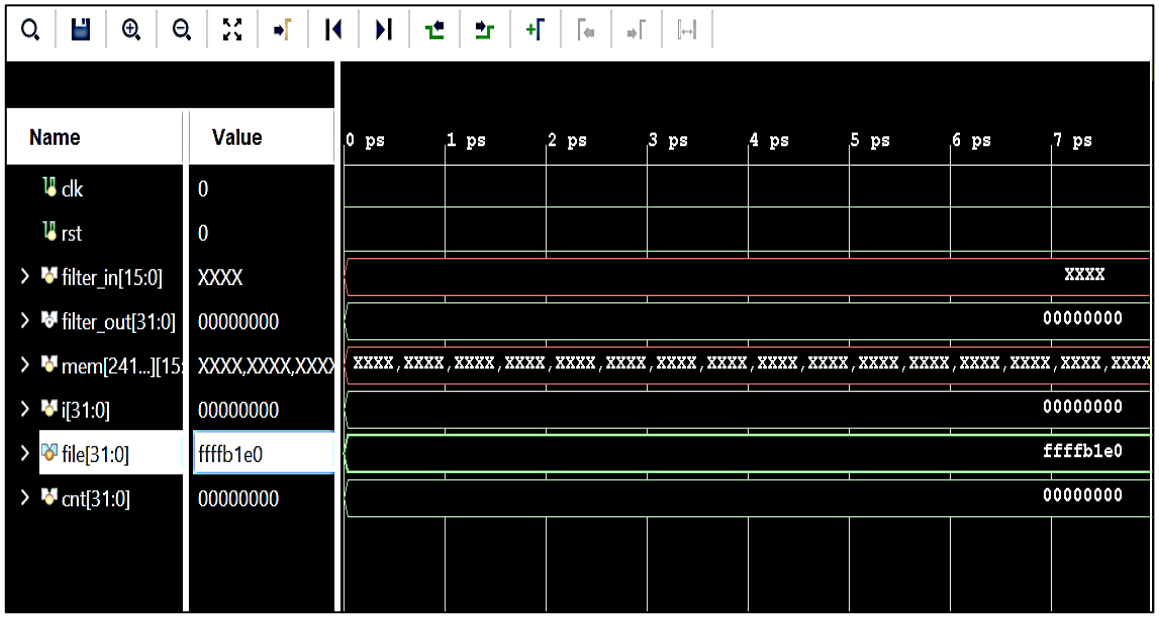


**Figure 6.** FIR Filter Behavioral Simulation

Figure 6, shows the behavioral design of the proposed FIR filter design with inputs and output

### C. FIR filter implemented design

FIR filter design implemented design simulated output is shown in figure 7.

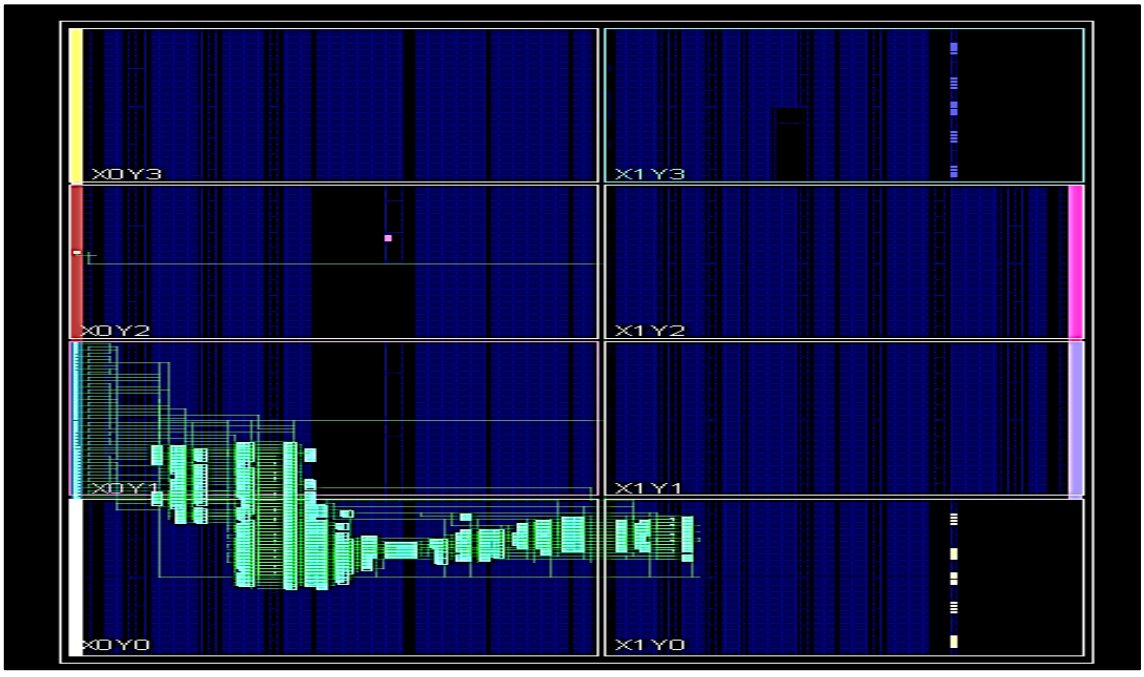


**Figure 7.** FIR filter implemented design

### D. FIR Filter LUT design runs

The synthesis and implemented completion report with usage of LUT, Flipflop, and DSP is shown in figure 8.



DOI: https://doi.org/10.54216/IJWAC.090206
Received: February 04, 2025 Revised: February 28, 2025 Accepted: March 30, 2025

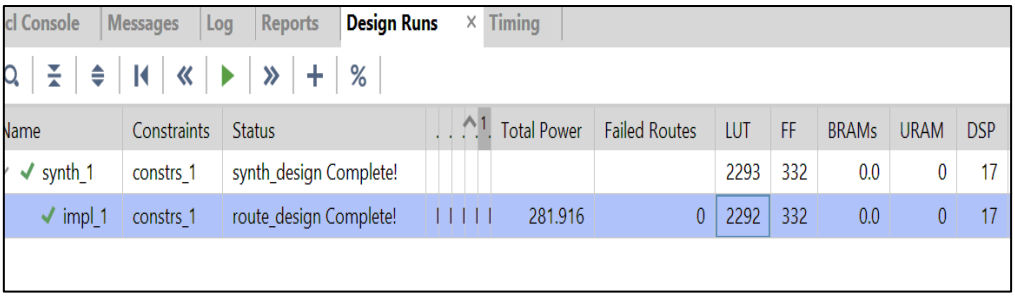


**Figure 8.** FIR Filter synthesis and implementation design runs

### E. Design ON-CHIP power summary

The Fig. 9 shows that on-chip power analysis and implemented in the netlist and dynamic on chip power is 281.916W

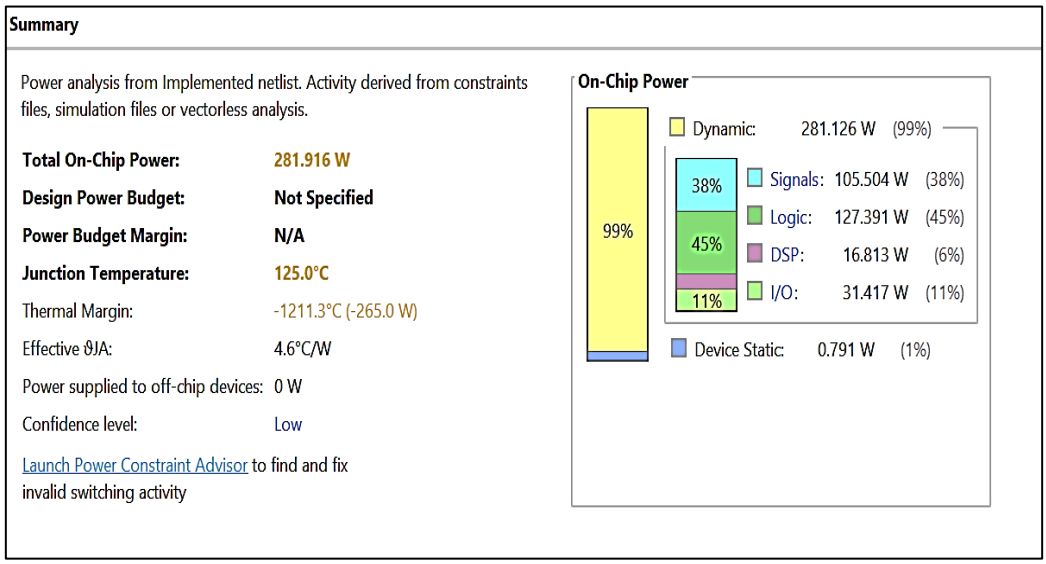


**Figure 9.** Summary of a design ON-CHIP Power

## 3. Conclusion

A basic analysis was conducted in terms of area, power, and delay. This multiplier based on the CLA adder proven to be considerably more efficient in terms of speed of operation than traditional multipliers. The synthesis findings demonstrate that the suggested FIR filter using modified multipliers that depends on CLA adders provides high speed while reducing hardware costs and power consumption when compared to standard FIR filters using other multipliers. In this future, the pipeline concept will be extended to include the adder unit used in digital FIR filters to achieve greater power and area savings. Furthermore, the design of a 16-tap FIR filter can be expanded to include n-taps for usage in real-time applications.

## References

[1] S. Janwadkar and R. Dhavse, "ASIC design of power and area efficient programmable FIR filter using optimized Urdhva-Tiryagbhyam Multiplier for impedance cardiography," *Microprocessors and Microsystems*, vol. 107, 105048, 2024, doi: 10.1016/j.micpro.2024.105048.

[2] M. Keerthi, V. Midasala, and S. N. Bhavanam, "FPGA implementation of distributed arithmetic for FIR filter," *Int. J. Eng. Res.*, vol. 1, no. 9, 2012.



DOI: https://doi.org/10.54216/IJWAC.090206
Received: February 04, 2025 Revised: February 28, 2025 Accepted: March 30, 2025

[3] Khan, A. Chakraborty, U. B. Joy, S. Wairya, and M. Hasan, "Carry look-ahead and ripple carry method based 4-bit carry generator circuit for implementing wide-word length adder," *Microelectronics J.*, vol. 140, 105949, 2023, doi: 10.1016/j.mejo.2023.105949.

[4] M. Kumm, A. Volkova, and S.-I. Filip, "Design of optimal multiplierless FIR filters with minimal number of adders," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 42, no. 2, pp. 658–671, 2023, doi: 10.1109/TCAD.2022.3179221.

[5] Z. Lin et al., "An incremental placement flow for advanced FPGAs with timing awareness," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 41, no. 9, pp. 3092–3103, 2022, doi: 10.1109/TCAD.2021.3120070.

[6] K. E. Murray et al., "Optimizing FPGA logic block architectures for arithmetic," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 28, no. 6, pp. 1378–1391, 2020, doi: 10.1109/TVLSI.2020.2965772.

[7] S. C. Prasanna, B. P. James, and V. Dhandapani, "High speed and area efficient coded input BCSM shared LUT-based FIR filter architecture," *Int. J. Syst. Assur. Eng. Manag.*, 2024, doi: 10.1007/s13198-024-02312-z.

[8] S. S. S. Priya, P. Karthigaikumar, and N. R. Teja, "FPGA implementation of AES algorithm for high-speed applications," *Analog Integr. Circuits Signal Process.*, vol. 112, no. 1, pp. 115–125, 2022, doi: 10.1007/s10470-021-01959-z.

[9] M. Raouf and S. Timarchi, "Non-volatile and high-performance cascadable spintronic full-adder with no sensitivity to input scheduling," *IEEE Trans. Circuits Syst. II: Express Briefs*, vol. 70, no. 6, pp. 2236–2240, 2023, doi: 10.1109/TCSII.2023.3237855.

[10] A. Share et al., "Design of high-speed BCD adder using CMOS technology," *IEEE Access*, vol. 11, pp. 141628–141639, 2023, doi: 10.1109/ACCESS.2023.3342328.

[11] P. Sivaprasad, V. Anandi, and S. Murthy, "Design of high throughput and low latency double precision floating point arithmetic unit for space signal applications," *Int. J. Comput. Sci. Netw. Secur.*, vol. 22, no. 9, pp. 808–815, 2022, doi: 10.22937/IJCSNS.2022.22.9.105.

[12] Z. Sun, Z. Zhou, and F.-W. Fu, "Optimizing code allocation for hybrid on-chip memory in IoT systems," *Integration*, vol. 97, 102195, 2024, doi: 10.1016/j.vlsi.2024.102195.

[13] L. Zhang, C. Rao, and X. Lou, "Low-power reconfigurable FIR filter design based on common operation sharing," *IEEE Trans. Circuits Syst. II: Express Briefs*, vol. 70, no. 8, pp. 3169–3173, 2023, doi: 10.1109/TCSII.2023.3256446.

[14] Kumar, R. K. Gupta, and M. K. Gupta, "Design of a high-speed and low-power 8-bit multiplier using Vedic mathematics," *Microelectronics J.*, vol. 118, pp. 105–112, 2022, doi: 10.1016/j.mejo.2022.105949.

[15] D. S. S. Raja et al., "A compact dual feed wide band slotted antenna for future wireless applications," *Analog Integr. Circuits Signal Process.*, vol. 118, pp. 291–305, 2024, doi: 10.1007/s10470-023-02233-0.

[16] S. Kumarganesh et al., "A novel analytical framework is developed for wireless heterogeneous networks for video streaming applications," *J. Math.*, vol. 2022, no. 1, pp. 1–7, 2022, doi: 10.1155/2022/2100883.



DOI: https://doi.org/10.54216/IJWAC.090206
Received: February 04, 2025 Revised: February 28, 2025 Accepted: March 30, 2025