EFFICIENT 3-PARALLEL POLYPHASE ODD LENGTH FIR FILTER USING KNOWLES ADDER AND COMPRESSOR BASED DADDA MULTIPLIER FOR VLSI APPLICATIONS

Tharigoppula Sushmitha[1], DR.T. Madhavi Kumari[2]

[1,2]Department of Electronics and Communication Engineering/Jawaharlal Nehru Technological University Hyderabad/India

Abstract

The demand for high-speed, low-power, and area-efficient hardware architectures has become essential in modern digital signal processing (DSP) systems and contemporary communication technologies. Finite Impulse Response (FIR) filters play a crucial role in these applications by ensuring stable and accurate signal manipulation, but their conventional designs often suffer from large hardware complexity, high propagation delay, and excessive power consumption. To address the drawbacks observed in earlier designs, this work proposes a high-performance 3-parallel polyphase FIR filter structure tailored specifically for odd-length filters. that integrates a Knowles adder and a compressor-based Dadda multiplier for optimized performance in Very Large Scale Integration (VLSI) applications. The design leverages a parallel polyphase structure to enhance throughput and computational speed, while the Dadda multiplier with compressor logic reduces the total count of intermediate partial-product compression levels, minimizing delay and switching power. The Knowles adder, with its balanced prefix tree structure, further improves speed and reduces interconnection complexity compared to conventional adders. The architecture introduced in this work is modeled in Verilog HDL, simulated and synthesized using Xilinx Vivado, and implemented on the Basys 3 FPGA board to validate its performance. Experimental results demonstrate that the developed FIR filter design achieves significant improvements in speed, area utilization, and power efficiency when evaluated against conventional FIR implementations that rely solely on basic multiplier–adder arrangements. The obtained outputs confirm accurate filtering operation, while synthesis results show reduced logic utilization and achieves a noticeably shorter critical path. Because of its improved computational performance, making the architecture highly appropriate for real-time DSP, biomedical processing, wireless communication, and other embedded VLSI applications.

Keywords: FIR Filter, Knowles Adder, Dadda Multiplier, Compressor, FPGA, VLSI Optimization, Parallel Processing.

I. Introduction

In modern digital systems, the demand for faster, energy-efficient, and compact hardware designs continues to increase with the evolution of communication and computing technologies. Digital Signal Processing (DSP) forms the backbone of these systems, with filtering operations serving as one of the most crucial components in noise reduction, signal extraction, and feature enhancement. Among various types of filters, Finite Impulse Response (FIR) filters are widely preferred for their stability, linear phase response, and predictable frequency characteristics. However, as signal complexity and data rates grow, conventional FIR filter designs face limitations in terms of processing speed, area efficiency, and power dissipation.

The performance of an FIR filter largely depends on the arithmetic units within it especially the adders and multipliers since they determine the system's overall delay and energy consumption. Traditional array multipliers and ripple-carry adders, although straightforward in design, are not well-suited for high-speed real-time processing due to their long propagation paths. This limitation has led researchers to explore parallel and pipelined designs that exploit concurrency and reduce computational latency. In particular, the polyphase decomposition technique has gained attention for allowing multiple filter sections to operate simultaneously, thus improving throughput without increasing hardware complexity.

Several researchers have contributed to optimizing FIR filter performance using advanced VLSI techniques. K. A. Rao et al. [1] proposed an efficient 3-parallel linear-phase FIR digital filter that reduced latency and improved data throughput by implementing a parallel processing architecture. Subsequently, M. Pandit and N. Purohit [2] advanced the earlier design by incorporating Brent–Kung adders along with Booth-encoded multipliers. This combination improved the overall computational speed and reduced power requirements, offering a more energy-efficient solution for high-performance filtering applications. Similarly, S. Chanda et al. [3] developed a 32-bit energy-efficient Dadda multiplier that effectively minimized power consumption while maintaining high-speed operation, which laid the foundation for future research in low-power arithmetic circuits.

In addition, researchers such as S.-F. Hsiao et al. [4] and Gu J. along with collaborators [5] focused on optimizing compressor architectures like 3:2, 4:2, and 5:2 compressors for faster partial product reduction. These works demonstrated that incorporating compressors significantly decreases propagation delay and power dissipation in multiplication operations. Meanwhile, hybrid adder designs like the Knowles adder, Kogge-Stone adder, and Han-Carlson adder have gained prominence for providing an ideal trade-off between area efficiency and speed. Among these, the Knowles adder offers balanced wiring complexity and excellent timing characteristics, making it highly suitable for high-performance VLSI systems.

Building upon these advancements, this work proposes an efficient 3-parallel polyphase odd-length FIR filter that integrates a Knowles adder with a compressor-based Dadda multiplier. The Dadda multiplier minimizes partial product levels through optimized compressor logic, while the Knowles adder accelerates final addition with low interconnect delay. This combination results in a filter design that offers improved computation speed, lower power consumption, and better area utilization compared to conventional FIR architectures.

The presented architecture has been implemented using Verilog HDL, simulated and synthesized on Xilinx Vivado, and implemented on the Basys 3 FPGA board. The obtained results indicate that the system delivers accurate filtering with high speed and reduced delay. Compared to traditional multiplier-adder combinations, the proposed architecture exhibits enhanced energy efficiency, making it suitable for real-time signal processing, biomedical applications, audio enhancement, and communication systems.

II. Literature Review

Multiplication and addition form the core arithmetic operations in modern digital signal processing, and they largely determine the performance of hardware circuits in terms of speed, area, and power. Over time, significant research has focused on refining these operations using optimized arithmetic structures. Among these, the Dadda multiplier and the Knowles parallel-prefix adder have shown strong potential for achieving fast and power-efficient computation in VLSI implementations. Continuous improvements in compressor-based reduction schemes and prefix adder networks have further strengthened these architectures, enabling higher computational throughput with reduced silicon overhead.

Chanda et al. (2019) proposed an energy-efficient 32-bit Dadda multiplier using optimized compressor structures to achieve high speed and low power. Their approach minimized partial product stages and effectively reduced switching activity, resulting in faster operation and reduced energy consumption. However, the design required additional logic resources for large bit-width applications, increasing the silicon area.

Rao et al. (2020) introduced a refined 3-parallel linear-phase FIR filter configuration that effectively handles odd-length coefficient sets. Their architecture utilized a polyphase decomposition technique to process multiple input samples simultaneously, improving throughput and performance. Although the design achieved significant speed enhancement, it required further optimization in arithmetic units to minimize power and delay.

Pandit and Purohit (2022) enhanced this concept by introducing a 3-parallel polyphase FIR filter that employed a Brent-Kung adder and Booth multiplier. Their architecture achieved lower power consumption and reduced delay compared to earlier designs. However, the design complexity increased due to multiple addition stages, indicating the need for a more compact and balanced structure.

Hsiao et al. (1998) proposed high-speed and low-power 3:2 and 4:2 compressor circuits for arithmetic operations, which minimized logic depth and improved propagation delay. Their design demonstrated the effectiveness of compressor-based reduction techniques in speeding up multipliers.

Gu. J et al. (2003) subsequently proposed a 4:2 compressor capable of operating reliably at very low voltages, making it highly suitable for battery-driven and power-restricted applications.While their design achieved excellent power efficiency, scalability remained limited for higher bit-width computations.

Reddy et al. (2019) implemented a 16-bit Wallace Tree multiplier that used 4:2 compressors combined with a Kogge-Stone adder for final summation. Their structure achieved higher speed compared to conventional multipliers, but the increased wiring and interconnect density led to higher power consumption. To address this, Patel et al. (2022) proposed an optimized Han-Carlson adder integrated with a tree-based multiplier structure. This adder demonstrated faster addition with lower area utilization, providing a better trade-off between speed and complexity.

Mishra et al. (2021) proposed a hybrid arithmetic structure combining a 5:2 compressor and Brent-Kung adder, emphasizing low power and compact design. Although their implementation achieved improved power efficiency, the delay remained relatively high due to the carry propagation path. Meanwhile, recent studies (2023–2024) have explored reconfigurable and approximate arithmetic circuits that dynamically balance speed and energy, targeting AI-integrated and adaptive digital hardware platforms.

A review of existing studies clearly shows that considerable progress has been achieved in the design and optimization of multiplier circuits and adder architectures for FIR filters. Yet, achieving an optimal combination of high speed, low power, and minimal area continues to be a challenge. This motivates the present work, which focuses on developing an efficient 3-parallel polyphase odd-length FIR filter using a Knowles adder and compressor-based Dadda multiplier. The The goal of the proposed architecture is to accomplish a balanced trade-off among performance parameters delivering faster computation, reduced power dissipation, and compact hardware utilization for real-time VLSI-based various real-time signal processing tasks.

III. Problem Statement

The efficiency of digital signal processing architectures largely depends on the efficiency of their arithmetic components, especially in Finite Impulse Response (FIR) filters. Traditional FIR filters using array multipliers and ripple-carry adders often suffer from high delay, large power consumption & increased area utilization. Existing architectures like Booth multipliers improve speed but introduce complex wiring and higher energy use. Therefore, there exists a strong demand for an optimized design that balances speed, energy efficiency and optimal hardware utilization. This work aims to develop an efficient 3-parallel polyphase odd-length FIR filter using a Knowles adder and compressor-based Dadda multiplier, providing high-speed performance with reduced delay, lower power, and compact hardware implementation suitable for VLSI applications.

IV. Methodology

The proposed design focuses on the implementation of an Efficient 3-Parallel Polyphase Odd-Length FIR Filter using a Knowles Adder and Compressor-based Dadda Multiplier in order to obtain high speed, low power consumption, and hardware compactness. Overall, the development methodology is structured into Three major stages:

1. Polyphase decomposition of the FIR filter,
2. Compressor-based Dadda multiplier design,
3. Knowles adder integration for fast addition.

A. Polyphase Decomposition of FIR Filter

A finite impulse response (FIR) filter represents a well-established DSP structure represents one of the most reliable and widely used architectures in digital signal processing operations. Its output is obtained by performing a discrete convolution between the incoming sample sequence and the corresponding filter coefficients. This mathematical operation can be expressed as:

$$y(n) = \sum_{k=0}^{N-1} \quad h(k) \cdot x(n-k)$$

Here, *N* denotes the total number of filter taps, *h(k)* indicates the corresponding impulse-response coefficient at index *k*, and *x(n − k)* denotes the value of the input sequence during sample index delayed by *k* time steps.

Although this direct form implementation is simple, it requires sequential multiplication and addition operations for each output sample, resulting in high computational latency for large filter orders.

To overcome these limitations, the filter is divided using a polyphase decomposition approach. The idea behind this method is to divide the filter into multiple smaller sub-filters, each responsible for processing a portion of the input samples. This allows for parallel computation, effectively reducing the overall delay and improving throughput.

In the proposed work, a 3-parallel polyphase decomposition is applied for an odd-length FIR filter. The sequence $x(n)$ represents the stream of input values supplied to the filter and is typically sampled in discrete time, which is divided into three parallel sub-sequences as follows:

$$x_0(n) = x(3n), x_1(n) = x(3n+1), x_2(n) = x(3n+2)$$

Similarly, the coefficients are divided into corresponding polyphase components:

$$h_0(n) = h(3n), h_1(n) = h(3n+1), h_2(n) = h(3n+2)$$

Each of these sub-filters operates independently on its respective data stream. The partial results from all three branches are then combined to reconstruct the final filter output, expressed as:

$$y(n) = y_0(n) + y_1(n) + y_2(n)$$

This decomposition enables the input sequence to be processed in three parallel paths, significantly lowering computational load per branch. As a result, the critical path becomes shorter, and the overall throughput of the filter increases.

The architectural layout of the developed 3-parallel polyphase odd-length FIR filter is illustrated in Fig. 1. Each branch includes delay units and multipliers associated with the respective filter coefficients, followed by an adder section that merges the intermediate outputs to form the final result.

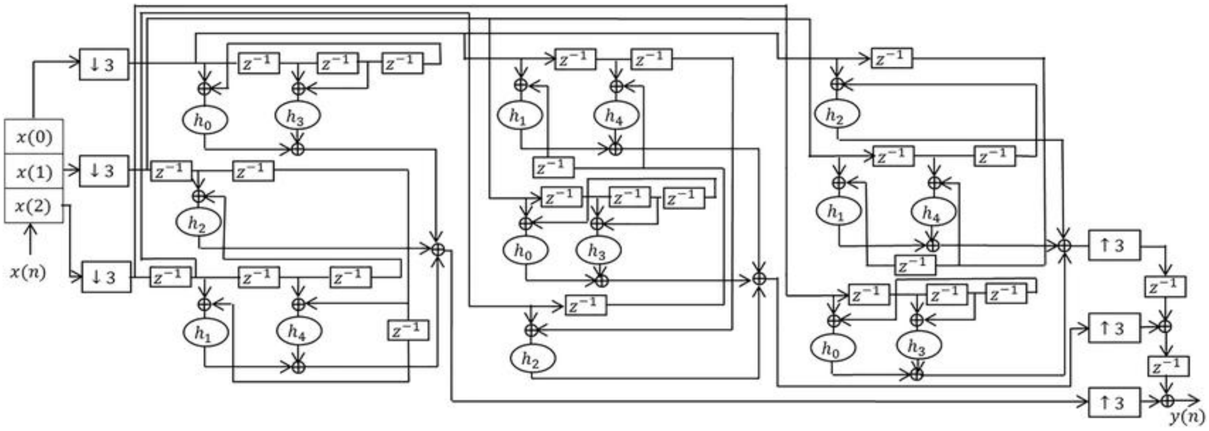


Fig.1. Polyphase Decomposition Structure of the 3-Parallel FIR Filter

This architecture achieves high-speed performance, reduced hardware utilization, and lower power dissipation, making it an efficient choice for real-time DSP and VLSI applications such as communication systems, biomedical signal analysis, and image enhancement.

B. Compressor-Based Dadda Multiplier

The multiplication process in FIR filters contributes the most to delay and power consumption. To optimize this stage, a Dadda multiplier is used instead of a conventional array multiplier.

The Dadda algorithm reduces partial products in a hierarchical manner using 4:2 and 5:2 compressors. These compressors perform bit-level reductions more efficiently than simple adders, thus shortening the critical path delay.

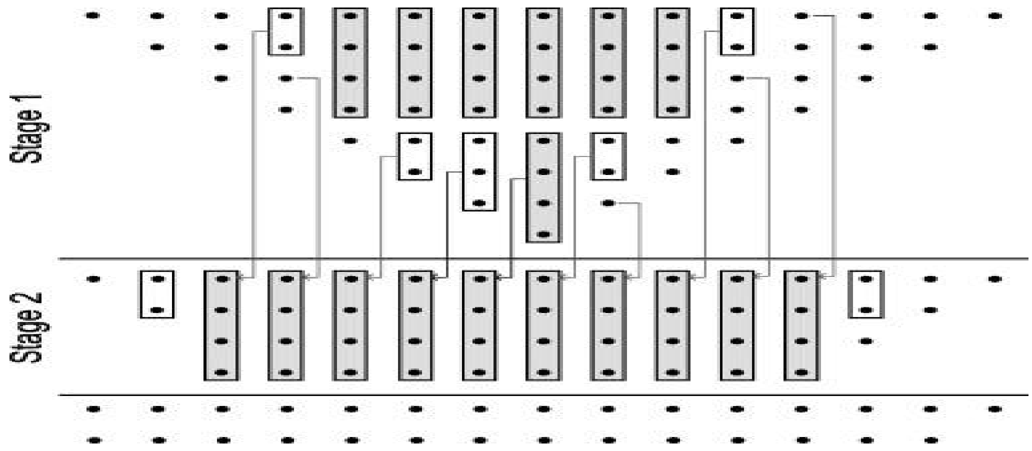


Fig.2 Block representation of the Dadda multiplier incorporating compressor-based partial product reduction.

Figure. 2 illustrates the architecture of the compressor-based Dadda multiplier embedded within the overall system design.

In the case of a multiplication of two binary numbers $A$and B, each of $n$bits:

$$P = A \times B = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} a_i b_j 2^{i+j}$$

Instead of adding all partial products sequentially, the Dadda structure compresses them in stages using 4:2 and 5:2 compressors.

- A 4:2 compressor takes 4 inputs and 2 carries, producing sum and carry outputs:

$$\text{Sum} = (A \oplus B) \oplus (C \oplus D)$$
$$\text{Carry} = (A \cdot B) + (C \cdot D)$$

- A 5:2 compressor further accelerates computation by handling five input bits and two carries simultaneously, producing three outputs: sum, carry, and carry-out.

The compressor-assisted Dadda multiplier reduces the number of internal addition layers, which effectively shortens the propagation time and helps lower the power requirements of the entire design.

C.Knowles Adder Design

After partial product reduction, the final summation is performed using a Knowles Adder, a type of parallel-prefix adder that provides an efficient balance between speed and area.

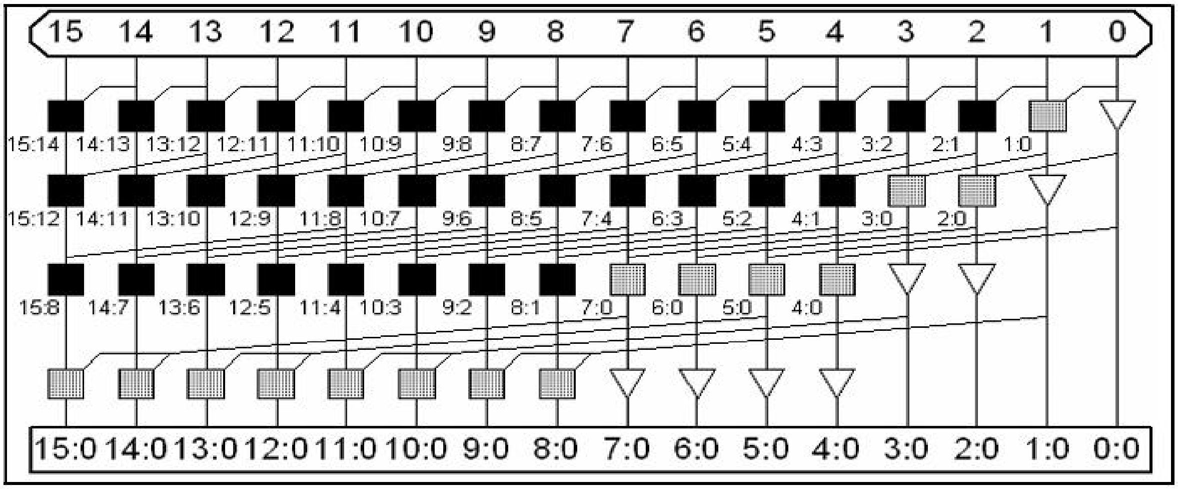


Figure 3. represents the Knowles Adder block structure, showing the carry-propagation network and the sum generation stage

Figure 3 presents the structural outline of the Knowles parallel-prefix adder employed in the final carry-propagation and sum-generation stage.

The Knowles Adder operates using the principles of generate (G) and propagate (P) signals defined as:

$$G_i = A_i \cdot B_i, P_i = A_i \oplus B_i$$

The carry generation in parallel-prefix adders follows the recursive relation:

$$C_{i+1} = G_i + (P_i \cdot C_i)$$

In the Knowles Adder, prefix computation is structured to balance fan-out and wiring complexity, ensuring high speed with minimal layout overhead.

Compared to other adders, the Knowles Adder achieves faster computation and reduced wiring complexity, making it suitable for FPGA-based implementations. Its balanced tree structure enhances scalability and reliability, which directly contributes to minimizing delay within the architecture associated with the developed FIR filter architecture.

V. RESULTS

This section presents the simulation, synthesis, and the FPGA-based implementation outcomes obtained for the proposed 3-parallel polyphase FIR filter structure. The system was implemented and validated through Verilog-based hardware modeling on the Xilinx Vivado Design Suite, and the hardware testing was carried out on the Basys3 FPGA board. The performance outcomes are evaluated primarily in terms of the end-to-end signal delay, power, and area and compared with the existing Design.

A. Simulation Result

The functional simulation of the designed 3-parallel polyphase FIR filter supporting odd-length coefficients was carried out using Xilinx Vivado to verify the correctness of the Verilog design before FPGA implementation. The top-level simulation waveform testbench is as as shown in Figure 4.

As observed in Figure 4, the system clock input (clk) and control input (BTNC) are implemented within the system to sequentially generate and observe outputs from the three parallel branches of the filter. The signals y0[7:0], y1[7:0], and y2[7:0] represent the three parallel output channels corresponding to different input and coefficient combinations.

At each positive clock edge, new input samples are processed through the compressor-based Dadda multiplier and Knowles adder, producing filtered output values in parallel. The waveform clearly shows the expected FIR output progression with values corresponding to input sets — (0, 10, 40), (80, 120, 160), and (140, 50, 0) — which match the theoretical calculations.

The LED [15:0] bus reflects the visual representation of two filter outputs, where the lower eight LEDs correspond to the first output and the upper eight LEDs display the second output. Meanwhile, the seg [6:0] and an [3:0] signals drive the seven-segment module, which shows the third parallel output in real time, allowing users to directly observe the filter behavior on the FPGA board.

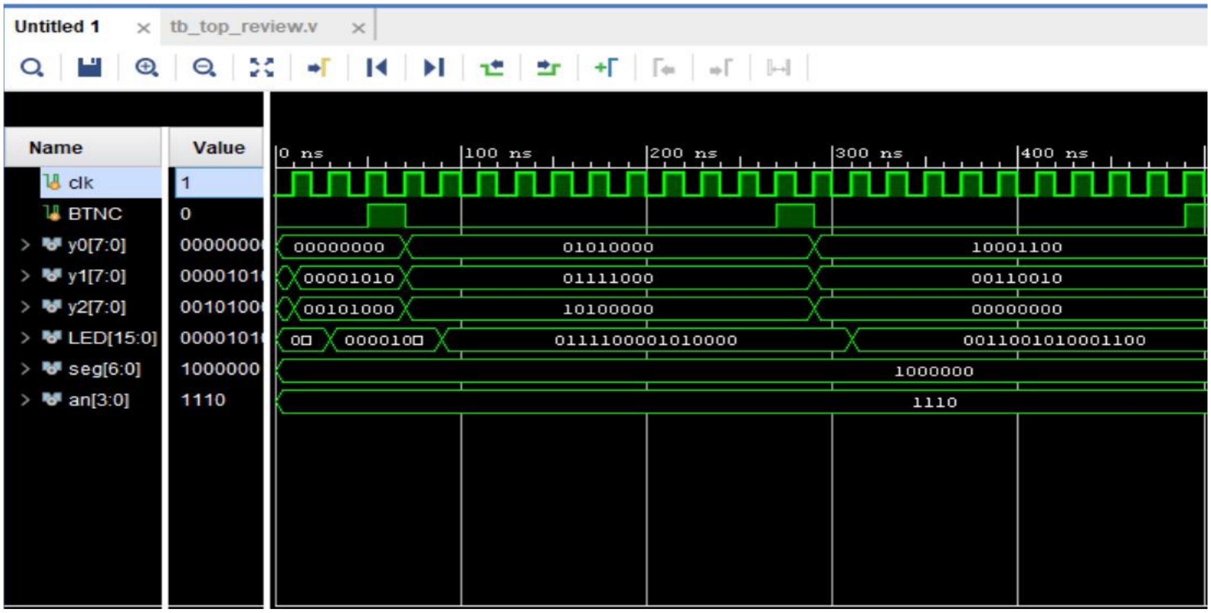


Figure 4. Simulation Waveform of the Proposed FIR Filter Showing Three Parallel Outputs

The uniform timing transitions between clock cycles confirm that the system achieves stable operation and accurate synchronization across all three filter branches. This verifies that the design successfully implements parallel FIR filtering with minimal delay and correct data propagation through each computation stage.

B. RTL Schematic Analysis

The RTL schematic of the proposed FIR filter, generated using Xilinx Vivado, provides a detailed view of the logical interconnections and data flow within the design.

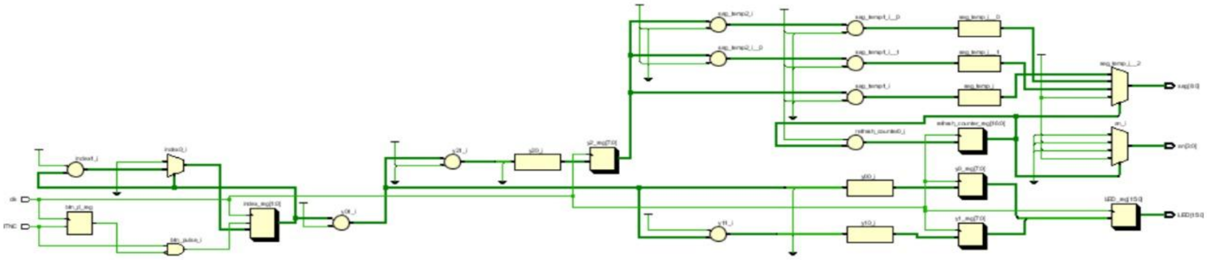


Fig.5 Synthesized RTL Schematic of the FIR Filter Architecture

It illustrates how various submodules including the compressor-driven Dadda multiplier, Knowles adder, and polyphase filter units are interconnected to perform parallel operations efficiently. The schematic confirms that data from input registers is distributed across three processing branches, ensuring true parallelism in computation. Multipliers and adders are efficiently mapped to FPGA logic resources, optimizing speed and reducing hardware redundancy. Overall, the RTL structure validates the correct hierarchical design and functional integration of all components before hardware synthesis.

C. FPGA Implementation

The proposed Efficient 3-Parallel Polyphase Odd-Length FIR Filter using Knowles Adder and Compressor-Based Dadda Multiplier was successfully implemented and tested on the Basys 3 FPGA board. The design was synthesized and downloaded onto the FPGA to validate its real-time functionality and verify the correctness of the simulation results. The FPGA implementation helps in demonstrating the practical feasibility of the proposed architecture, confirming its efficiency in terms of performance, resource utilization, and accuracy.

In the hardware setup, the first eight LEDs (LD0–LD7) of the Basys 3 board were assigned to display the first output (y_0), while the remaining eight LEDs (LD8–LD15) were used to indicate the second output (y_1). The 7-segment display was employed to display the third output value (y_2), allowing real-time observation of all three parallel outputs produced by the filter.

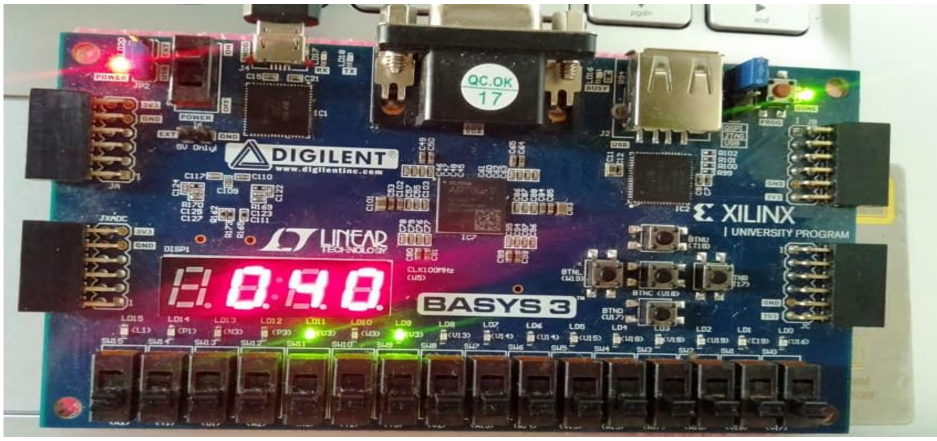


Fig.6(a)

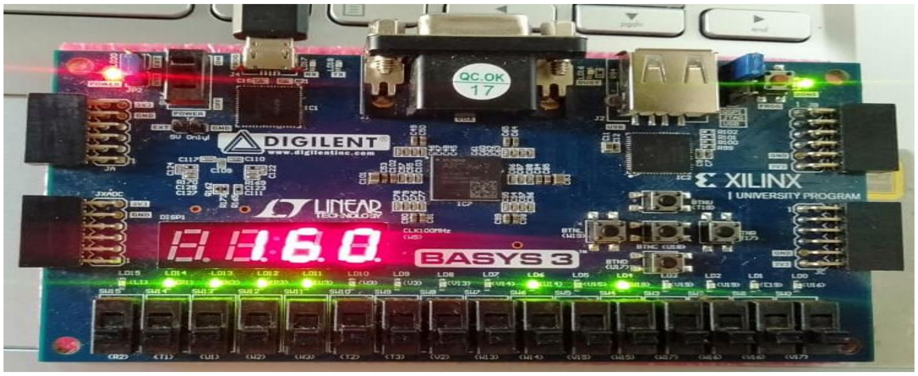


Fig.6(b)

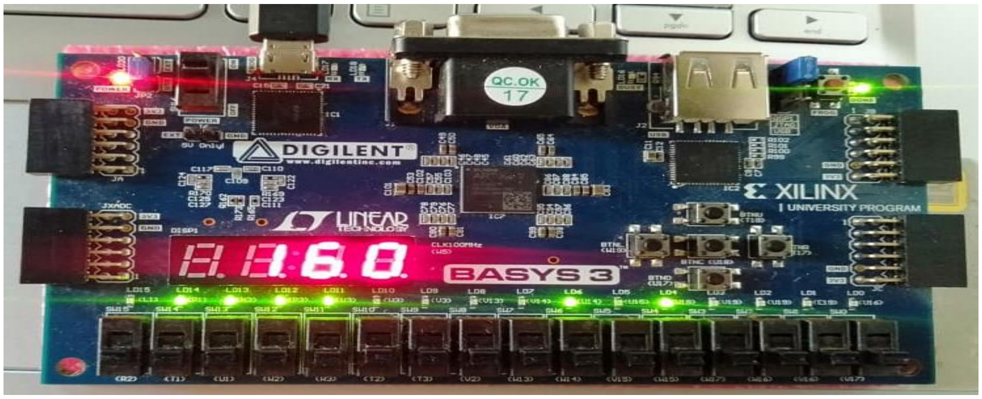


Fig.6(c)

Fig.6 FPGA Implementation of proposed 3-Parallel Polyphase Odd-Length FIR Filter using Knowles Adder and Compressor-Based Dadda Multiplier

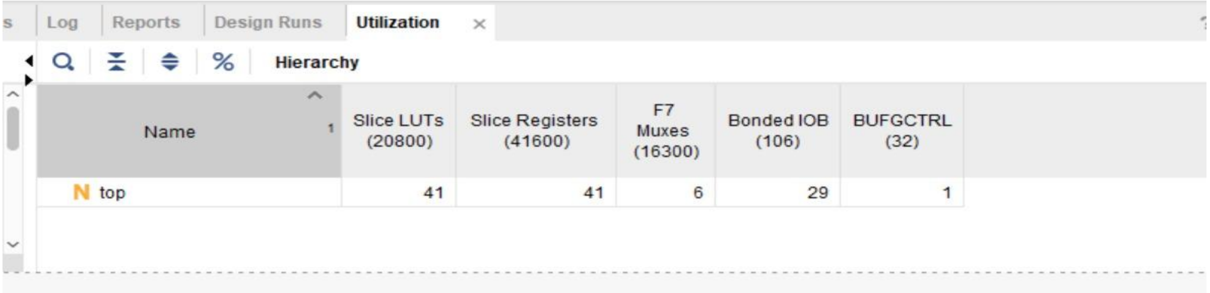
Figure 6(a) presents the first set of filter responses, where the LEDs and seven-segment interface display the outputs (0, 10, 40). To display the third output value (y_2), allowing real-time observation of all three parallel outputs produced by the filter. Figure 6(b) illustrates the second group of outputs (80, 120, 160), representing the steady-state functional behavior of the filter where all input samples and coefficients are fully active, producing the maximum output levels. Finally, Figure 6(c) depicts the third set of results (140, 50, 0), capturing the tail-end of the filtering process as the input samples move through the processing chain, causing the resulting output magnitudes to gradually decrease.

The results observed on the FPGA matched the simulated waveform results precisely, validating the correctness of the filter design. The uniform transition of LED patterns combined with accurate numerical updates on the seven-segment display clearly demonstrate the fast response time and reliable hardware behavior achieved by the proposed filter design. The FPGA-based realization thus demonstrates that the implemented FIR filter achieves the desired performance goals in terms of speed, low power, and area efficiency, making it suitable for high-speed DSP and VLSI applications.

D. Synthesis Results

The proposed Efficient 3-Parallel Polyphase Odd-Length FIR Filter using Knowles Adder and Compressor-Based Dadda Multiplier was synthesized using Xilinx Vivado targeting the Basys 3 FPGA (XC7A35T) device. The synthesis reports provide detailed insights into the hardware resource utilization, power estimation, and timing performance, which together validate the design efficiency in terms of speed, area, and energy consumption.

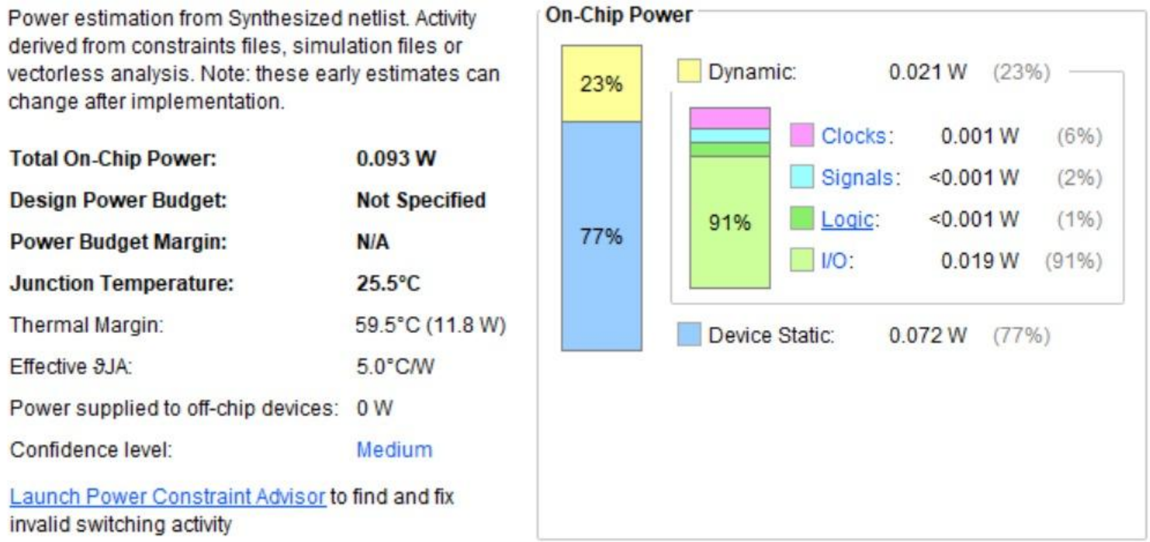
Figure 7 presents the area utilization summary, detailing the hardware resources utilized by the proposed design. The architecture occupies 41 Slice LUTs, 41 Slice Registers, 6 F7 multiplexers, and 29 I/O blocks, demonstrating a compact implementation on the FPGA. This low resource usage is mainly due to the optimized use of compressor-based reduction and the efficient structure of the Knowles adder. The reduced LUT count reflects the design's effectiveness in minimizing hardware overhead compared to conventional FIR filter realizations.



Name	Slice LUTs (20800)	Slice Registers (41600)	F7 Muxes (16300)	Bonded IOB (106)	BUFGCTRL (32)
N top	41	41	6	29	1

Fig.7 Area Utilization Report

The power analysis report in Figure.8 shows an overall device power usage measured at 0.093 watts, with 0.072 watts as static power and 0.021 W as dynamic power. The dynamic power primarily arises from I/O operations, while the logic and clock networks contribute minimally. The low overall power value demonstrates that the developed architecture functions efficiently with minimal energy dissipation, making it highly suitable for portable and real-time VLSI-based DSP systems.



Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power:	**0.093 W**
Design Power Budget:	**Not Specified**
Power Budget Margin:	**N/A**
Junction Temperature:	**25.5°C**
Thermal Margin:	59.5°C (11.8 W)
Effective ϑJA:	5.0°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Medium

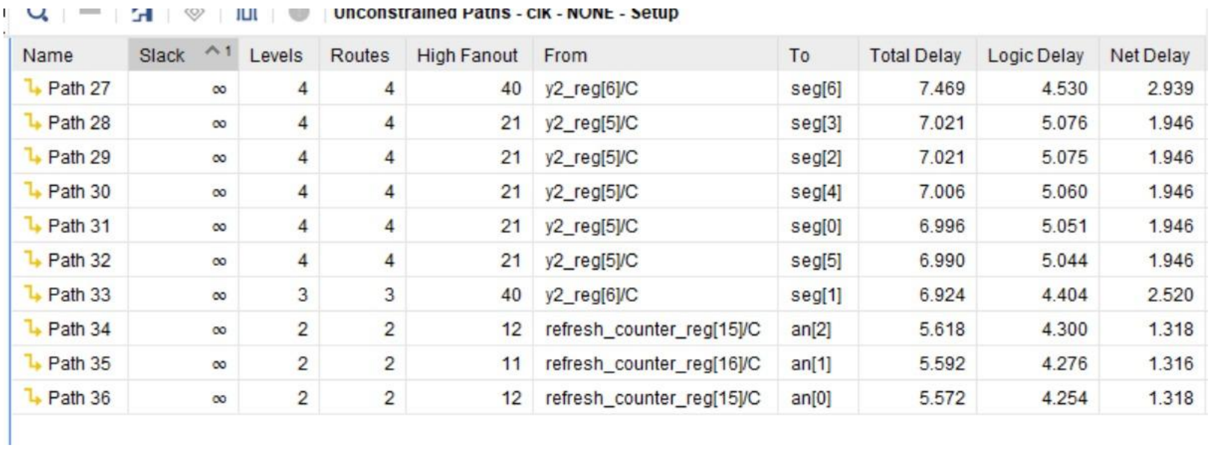
Launch Power Constraint Advisor to find and fix invalid switching activity

On-Chip Power

Dynamic:	0.021 W	(23%)
Clocks:	0.001 W	(6%)
Signals:	<0.001 W	(2%)
Logic:	<0.001 W	(1%)
I/O:	0.019 W	(91%)
Device Static:	0.072 W	(77%)

Fig.8.Power Report

The timing summary, presented in Figure.9 indicates a maximum delay of approximately 7.46 Nanoseconds, which aligns with the maximum combinational path between input and output signals. This reduced delay highlights the effectiveness of the compressor-based Dadda multiplier and Knowles adder, both optimized for high-speed computation. The reduced propagation delay ensures faster data throughput and improved operational speed, validating the performance enhancement achieved through architectural optimization.



Unconstrained Paths - clk - NONE - Setup

Name	Slack ^1	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay
Path 27	∞	4	4	40	y2_reg[6]/C	seg[6]	7.469	4.530	2.939
Path 28	∞	4	4	21	y2_reg[5]/C	seg[3]	7.021	5.076	1.946
Path 29	∞	4	4	21	y2_reg[5]/C	seg[2]	7.021	5.075	1.946
Path 30	∞	4	4	21	y2_reg[5]/C	seg[4]	7.006	5.060	1.946
Path 31	∞	4	4	21	y2_reg[5]/C	seg[0]	6.996	5.051	1.946
Path 32	∞	4	4	21	y2_reg[5]/C	seg[5]	6.990	5.044	1.946
Path 33	∞	3	3	40	y2_reg[6]/C	seg[1]	6.924	4.404	2.520
Path 34	∞	2	2	12	refresh_counter_reg[15]/C	an[2]	5.618	4.300	1.318
Path 35	∞	2	2	11	refresh_counter_reg[16]/C	an[1]	5.592	4.276	1.316
Path 36	∞	2	2	12	refresh_counter_reg[15]/C	an[0]	5.572	4.254	1.318

Fig.9. Timing and Delay Report

Overall, the synthesis results confirm that the designed FIR filter architecture attains a strong balance between delay, power usage & hardware area. The integration of the employed techniques compressor-based Dadda multiplier and Knowles adder enables a high-performance, energy-efficient solution suitable for real-time and resource-constrained DSP applications.

E. Performance Analysis

A comparative Analysis was conducted between the existing Conventional FIR Filter, the Brent-Kung Adder with Booth Multiplier FIR Filter, and the Proposed Knowles Adder with Compressor-Based Dadda Multiplier FIR Filter. The comparison focuses on three major parameters: area utilization (LUTs), power consumption (W), and propagation delay (ns) which are critical indicators of hardware efficiency and overall design performance.

Parameter	**Conventional FIR Filter**	**Brent-Kung + Booth Multiplier FIR Filter**	**Proposed Knowles + Dadda FIR Filter**
Area (LUTs)	68	59	41
Power (W)	0.136	0.110	0.093
Delay (ns)	9.12	8.13	7.46

Table.1 Comparison Table

The comparative data clearly reveals that the proposed FIR filter achieves superior performance across all measured parameters. The area utilization has been reduced to 41 LUTs, compared to 68 in the conventional and 59 in the Brent-Kung–Booth design, showing efficient hardware optimization through the Knowles adder and compressor-based Dadda multiplier. The system's power requirement is further brought down to 0.093 W, lower than both existing designs, mainly due to reduced switching activity and improved logic efficiency.

Furthermore, the signal delay is lowered to 7.46 nanoseconds, indicating faster computation compared to 9.21 ns and 8.13 nanoseconds in the other two designs. This improvement reflects the high-speed carry propagation and balanced structure of the Knowles adder.

Overall, the proposed design achieves a balanced compromise between high processing speed, power, and area, making it highly suitable for real-time digital signal processing and low-power VLSI applications.

VI. CONCLUSION

The proposed Efficient 3-Parallel Polyphase Odd-Length FIR Filter utilizing a Knowles Adder and Compressor-Based Dadda Multiplier has been effectively designed and implemented to achieve enhanced performance in terms of speed, power, and area. Through simulation and FPGA realization on the Basys 3 board, the system demonstrated accurate functional behavior and stable hardware operation. The optimized combination of the Dadda multiplier and 4:2 compressor greatly minimized the number of intermediate partial-product compression levels, minimizing propagation delay. Meanwhile, the Knowles adder provided faster carry computation with reduced wiring complexity, contributing to a more area-efficient and high-speed filter architecture.

The experimental analysis revealed notable improvements over existing FIR filter designs. The proposed architecture demonstrates notable improvements across all performance metrics. It requires only 41 LUTs, leading to a smaller hardware footprint and using a low power of 0.093 W and achieving a minimized delay of 7.46 ns. These results show superior efficiency compared with conventional FIR filters and those based on Brent–Kung and Booth multipliers. Such enhancements make the design highly suitable for real-time DSP applications where rapid computation and low-energy operation are essential. The architecture may be further extended in the future to support higher-order filters, adaptive systems, or reconfigurable FPGA-based DSP platforms, offering broader scalability and enhanced functionality.

REFERENCES

[1] S.-F. Hsiao, M.-R. Chen, and C.-T. Hong, “High-speed and low-power 3-2 counter and 4-2 compressor for fast multipliers” Electronics Letters, volume 34, number 4, pp. 341–343, 1998.

[2] Gu. J and Chang C. H. “Ultra low-voltage and low-power 4–2 compressor architecture for high-speed arithmetic operations,” IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, volume 54, Issue 5, pages 412–416, published in May 2003.

[3] B. Reddy and V. Kumar, “Design of high-speed Wallace tree multiplier using 4-2 compressor and Kogge-Stone adder” International Journal of Engineering Research & Technology, volume 8, number 5, pp. 205–210, May 2019.

[4] P. Singh and R. Kumar, “High-speed multiplier design using 5-2 compressors,” International Conference on Communication and Signal Processing, Pages 654–659, released in the year 2020.

[5] N. Mishra, P. Patel, and M. Chauhan, “Hybrid multiplier using 5:2 compressor and Brent-Kung adder for efficient VLSI implementation” IEEE Access, volume 9, pp. 15847–15856, 2021.

[6] J. Patel and H. Shah, “Design of high-speed Han-Carlson adder integrated with Wallace tree structure,” International Journal of Innovative Academic research Journal of Computer and Communication Engineering, Volume 9, no. 7, pp. 1211–1218, July 2022.

[7] S. Chanda, K. Guha, S. Patra, A. Karmakar, L. M. Singh, and K. L. Baishnab, “Design of an energy-efficient exact 32-bit Dadda multiplier,” presented at the IEEE Fifth International Conference on Convergence at the I2CT conference held in Mumbai, India, in 2019, spanning pages 1–4.

[8] K. A. Rao, A. Kumar, and N. Purohit, “Efficient implementation for 3-parallel linear-phase FIR digital odd length filters,” 2020 IEEE 4th Information and Communication Technology Conference (CICT), held in Chennai, India, 2020, pp. 1–6.

[9] Rao, K. A., and Pandit, M, and N. Purohit, *“*A 3-parallel polyphase odd-length FIR filter employing Brent–Kung adders and Booth multipliers for VLSI systems,” in the proceedings of the Ninth IEEE Uttar Pradesh Section Conference on Electrical Engineering, Electronics and Computer Engineering (UPCON), Prayagraj, India, 2022, pp. 1–5.

[10] S. T. Bala, M. K. Bansal, and P. Saini introduced an approximate Wallace tree multiplier that incorporates a 4:2 compressor-driven reduction structure. Their work appeared in the International Journal of Engineering and Advanced Technology (IJEAT), Volume 8, Issue 5, pages 1543–1548, published in 2019.

[11] P. D. H. Knowles, “A family of adders,” Published in the IEEE Symposium proceedings on Computer Arithmetic, pp. 277–281, published in the year 1991.

[12] Wallace, C. S, “Design of a high-performance multiplier framework,” reported in IEEE Transactions on Electronic Computers, Volume EC-13, Issue 1, pages 14 through 17, issued in February 1964.

[13] L. Dadda, “Some schemes for parallel multipliers,” Alta Frequenza, volume 34, pages 349–356, published in 1965.

[14] B. Parhami, Computer Arithmetic: Algorithms and Hardware Designs, 2nd edition, Oxford University Press, 2010.

[15] Mitra, S. K., Digital Signal Processing: A Computational Perspective, Fourth Edition, McGraw-Hill, 2010.

[16] D. S. Prasad and P. R. Kumar, “FPGA implementation of high-performance FIR filter using parallel processing” Procedia Computer Science, volume 143, pp. 573–580, 2018.

[17] A. R. Akula and S. B. P., “FPGA-based implementation of 3-parallel polyphase FIR filter using Dadda multiplier,” International Journal of Advanced Scholarly studies in Electronics and Communication Engineering, volume 10, number 4, pp. 445–451, 2021.

[18] R. B. Choudhary and N. Patel, “High-speed digital filter design using compressor and hybrid adder,” IEEE International Conference on Communication and Signal Processing (ICCSP), pp. pages 654 through 659, April 2022.

[19] S. R. Prakash and G. Srinivas, “Hardware optimization of FIR filter using parallel and polyphase structures” IEEE Transactions on Circuits & Systems, Series II volume 69, number 8, pp. 3241–3250, August 2022.

[20] V. K. Sharma, “Low-power design of DSP processors using compressor-based Dadda multipliers,” Volume 15 of the “International Journal of Electronics and Communication Engineering”, number 5, pp. 305–312, 2023.