# Efficient Implementation of FIR Filter Utilizing Dadda Multiplier and Brent Kung Adder

M. Deepika Krishna[1*], V. Siva Naga Hanusha[2], V. Vinay Guru Charan[2], Sk. Dariyavali[2],
Syed. Md. Ameer[2]
[1]Assistant Professor, [2]Under Graduate Student, Department of Electronics Communication Engineering, Seshadri Rao Gudlavalleru Engineering College, Vijayawada, Andhra Pradesh, India
[*]Corresponding Author: deepikakrishna5555@gmail.com

*Received Date: May 15, 2024; Published Date: May 15, 2024*

***Abstract***
*In signal processing and communication systems, digital filters play a pivotal role, as they are one of the fundamental components of traditional DSP processors. The performance of digital filter multipliers and adders stands out, as they are essential for carrying out various mathematical operations. However, multipliers often dominate in terms of silicon space and power consumption, and adders incur delays. A novel technique for developing efficient FIR filters must be proposed to address this challenge. The primary objective is to achieve an optimal trade-off between power consumption and computational efficiency. By leveraging innovative design strategies, such as the Dadda multiplier, which enables concurrent processing of multiple bits and the Brent-Kung adder, which employs a car look ahead strategy to minimize critical path delays and diminish power consumption, the performance of FIR filters can be enhanced in signal processing applications compared to regular adders. The proposed approach involves utilizing a Dadda multiplier and a Brent-Kung adder for the final stages of merging in the filter design. This combination aims to strike a balance between computational efficiency and power usage. To evaluate the system's effectiveness, it is implemented in Verilog HDL and compared against a Booth-encoded multiplier. This work underscores the importance of efficient hardware implementation in achieving high-performance digital signal processing systems.*

**Keywords-** Brent Kung Adder, Communication, Dadda multiplier, FIR filter, Digital Signal Processors (DSPs)

## INTRODUCTION

In modern signal processing and communication systems, digital filters are indispensable components, playing a pivotal role in myriad applications, from audio processing to telecommunications [1]. These filters are instrumental in extracting meaningful information from noisy signals, equalizing frequency responses, and separating desired signals from interference. At the core of such systems lie Digital Signal Processors (DSPs), which execute intricate mathematical algorithms to process signals in real-time [2]. Among the critical building blocks of these algorithms are multipliers and adders, essential for performing arithmetic computations fundamental to filtering operations.

Despite their significance, the relentless pursuit of higher performance in DSP systems presents formidable challenges regarding silicon area and power consumption [3]. Conventional multiplier designs, in particular, have been known to impose substantial overhead on chip real estate and power budgets. Consequently, there is a pressing need for innovative solutions that can reconcile these efficiency concerns while preserving or even enhancing system performance. Addressing these challenges head-on, this project proposes a pioneering approach to digital filter implementation by harnessing advanced multiplier and adder architectures for VLSI applications [4]. By incorporating efficient designs such as the Dadda multiplier and the Brent-Kung adder, the proposed methodology aims to optimize silicon area utilization and power consumption [5]. The Dadda multiplier, renowned for its parallel architecture and exceptional area efficiency, enables concurrent processing of multiple bits, thereby bolstering overall throughput [6]. Conversely, the Brent-Kung adder employs a carrlookaheadad strategy to minimize critical path delays and diminish power consumption, further amplifying efficiency gains.

The proposed implementation methodology strives to meet the evolving demands of modern DSP applications through the seamless integration of these cutting-edge architectures. By fine-tuning resource allocation and fortifying computational efficiency, the approach endeavours to satisfy the escalating requirements for high-performance digital filtering systems [1]. Ultimately, this project represents a significant stride towards realizing more resource-efficient and potent signal processing solutions capable of catering to various applications across various domains.

Moreover, adopting the Dadda multiplier and the Brent-Kung adder enhances efficiency and fosters scalability and versatility in digital filter design [7]. The modular nature of these architectures facilitates easy integration into larger DSP systems, allowing for seamless expansion and adaptation to evolving requirements. This scalability is particularly advantageous in scenarios where the complexity of filtering algorithms may vary, or additional functionality must be incorporated into the system without compromising performance.

Furthermore, utilizing innovative multiplier and adder designs underscores the importance of holistic optimization strategies in DSP system design [8]. Beyond individual component performance, achieving optimal efficiency necessitates a comprehensive approach considering the interplay between various system elements. By meticulously optimizing the arithmetic units and the overall system architecture, this project exemplifies the potential for synergistic gains in performance, power efficiency, and resource utilization [2]. Through continued exploration and refinement of such holistic design methodologies, digital signal processing is poised to unlock new frontiers of capability and efficiency, empowering transformative advancements across diverse domains and applications [9].

## LITERATURE SURVEY

K. A. Rao [5], etc. all, the paper explores DSP processor optimization, focusing on reducing power consumption and improving area efficiency, particularly in Parallel Finite Impulse Response (FIR) filters. It introduces FFA-based 3-parallel polyphase FIR filters with optimized adders and multipliers, highlighting comparisons with traditional components. Tests on FPGA Artix-7 with Xilinx Vivado show improvements in delay and area, emphasizing the suitability of Booth multiplier and Brent Kung adder for low-power and compact VLSI applications.

Andrea Bonetti [6], etc. all, this overview thoroughly examines FIR filter multipliers, extensively analyzed through power simulations. It introduces a method for adjusting the coefficients of baseline filters at the algorithmic level to balance reduced power consumption against filter quality. Notably, the proposed optimization technique is free of any additional hardware resources. It also allows for the dynamic scaling of filter power consumption during operation while maintaining the complete baseline performance of any programmed filter as needed.

R. Hedge [1], etc., a low-power digital filter has been developed using a 0.35-micrometer 3.3-volt CMOS process. Remarkable energy efficiency is achieved beyond what's possible with voltage scaling alone by combining Voltage over Scaling (VOS) and algorithmic noise tolerance (ANT). The approach involves pushing the supply voltage beyond typical limits (VOS) and implementing ANT to mitigate performance degradation. Results indicate a substantial reduction in energy dissipation (40%-67%) compared to optimized voltage scaling, with minimal loss in signal-to-noise ratio (SNR) across a wide range of filter bandwidths (0.05fsub s-0.25fsub s, where fsub s is the sampling frequency).

P. N. Whatmough [3], etc. all, this paper introduces a novel technique to enhance energy efficiency in digital signal processing circuits by addressing timing errors without sacrificing robustness. It employs razor flip-flops to detect timing errors and dynamically adjusts voltage scaling. Instead of traditional error correction, it limits timing errors by creating a timing guard band, ensuring critical paths correspond to the least significant bit registers.

## FIR FILTER

FIR filters, also known as finite impulse response filters, are one of the most critical components in digital signal processing. They are employed for various purposes, including noise reduction, equalization, and signal separation. However, in contrast to infinite impulse response (IIR) filters, finite impulse response (FIR) filters have a finite impulse response, meaning that the filter's output response settles to zero in a finite number of samples after an input signal is applied. This characteristic inherently lends stability to FIR filters because they do not have feedback loops that can potentially introduce instability, making them more stable and providing linear phase characteristics, which means that all frequencies within the pass band are delayed by the same amount of time. Linear phase is desirable in many applications because it preserves the input signal's waveform shape and avoids distortion caused by phase shifts. Unlike IIR filters, FIR filters do not have pole-zero placement constraints. This lack of constraints allows for a more straightforward implementation of linear phase characteristics without worrying about stability issues associated with pole locations. When the input is a unit impulse function, the output of a FIR filter is represented by the filter's impulse response characteristics.

At the time index (n), the output $y(n)$ of a Fourier transform (FIR) filter may be defined as the convolution sum of the filter coefficients $h(k)$ and the input signal $x(n-k)$:

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

$y(n)$ is a non-negative number. Where:

- The number of taps or filter coefficients is denoted by the symbol $N$.
- The filter coefficients are denoted by the symbol $h(k)$.
- A delay of k samples is denoted by the symbol $x(n-k)$, which represents the input signal.

The filter coefficients of an FIR filter define the filter's impulse response. These coefficients are set based on the frequency response characteristics that are specifically wanted. Various windowing methods, including Hamming, Hanning, and Kaiser, are standard practices for designing FIR filters with specific frequency responses.

The transfer function $H(z)$ of a Fourier transform (FIR) filter in the z-domain may be expressed as follows:

$$\text{H(Z)}=\sum_{k=0}^{N-1} h(k).\, z^{-k} \quad (2)$$

Where $z$ is the complex variable representing the Z-transform of the input and output signals.
By substituting $z$ with $e^{j\omega}$, where $\omega$ represents the digital frequency covering the range from $-\pi$ to $\pi$, it is possible to derive the frequency response of a Fourier transform infrared (FIR) filter. The frequency response, denoted by the symbol $H(e^{j\omega})$, is a function that represents the amplitude and phase properties of the filter at various frequencies.

$$\text{H}(e^{j\omega})=\sum_{k=0}^{N-1} h(k).\, e^{-j\omega k} \quad (3)$$

When designing FIR filters, it is necessary to pick the correct filter coefficients, $h(k)$, to produce the required frequency response characteristics. These characteristics may include low-pass, high-pass, band-pass, or band-stop features. In signal processing, it is standard practice to use optimization techniques like the Parks-McClellan algorithm or the frequency sampling method when designing FIR filters. These approaches are utilized to suit particular needs in signal processing applications.

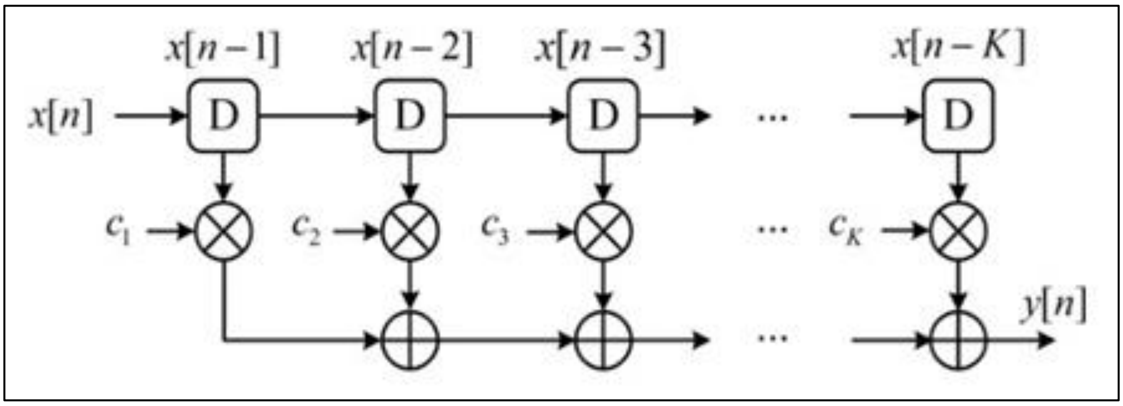

***Figure 1:*** *Block diagram of FIR filter.*

Fig. 1 shows the block diagram of a FIR (Finite Impulse Response) filter, which can be described as, overall, the FIR filter operates by delaying the input signal, multiplying the delayed samples by coefficients, and summing the results to produce the filtered output signal.

## DADDA MULTIPLIER

The Dadda multiplier is a multiplier design frequently utilized in digital signal processing applications. It is characterized by its fast speed and effective use of space. Reducing the number of partial product rows and optimizing the creation of partial products is particularly well-suited for efficiently multiplying large operands.

First, let us define two binary numbers, *A* and *B*, to be multiplied. Let us assume that *A* possesses *M* bits, whereas *B* possesses *N* bits. The functioning of the Dadda multiplier involves breaking down the multiplication operation into partial products and then effectively combining those products to produce the maximum possible output.

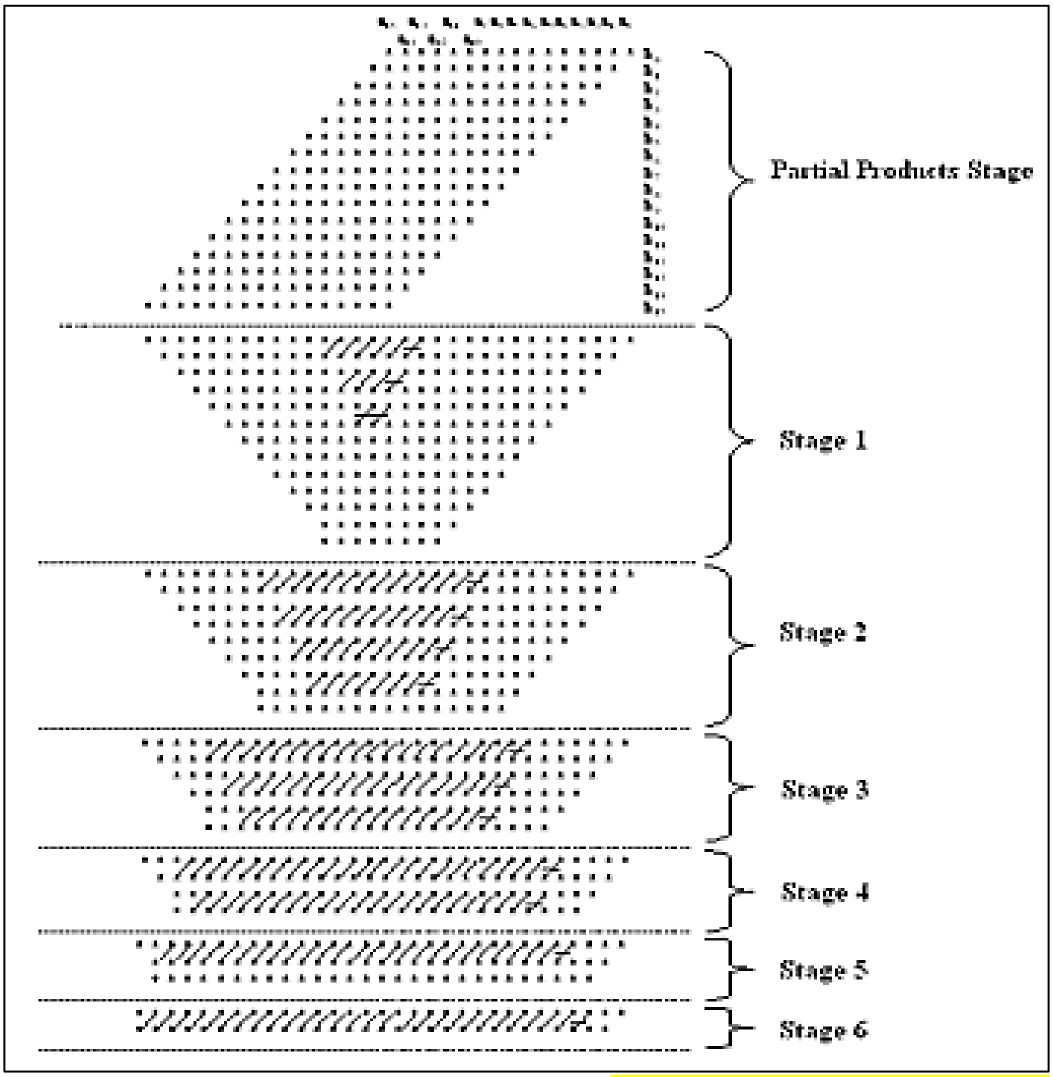

***Figure 2:*** *Flow representation of Dadda Multiplier. Please provide clear image if possible*

Fig. 2 shows the flow representation of the Dadda multiplier, which typically involves the production of partial products accomplished by processing shifting and adding operations. A partial product is produced for each bit in multiplier *B* by moving the multiplicand *A* to the relevant bit position and carrying out a bitwise AND operation. This process is repeated for each bit in the multiplier. Rows organize these partial products according to the bit locations in which they occur.

The Dadda compression tree effectively

combines these partial products by repeatedly decreasing the number of rows until only two are left. The tree removes superfluous partial products and improves the hardware usage at each level of the compression process. Because of this compression process, the number of partial products that need to be processed in parallel is significantly reduced, resulting in a design that can better use available space.

The final result is reached by employing a fast adder architecture to summate the partial products that have been blended. The Dadda multiplier design often uses high-speed adders for quick and practical addition. Some examples of these adders are the Brent-Kung adder and the CarrLookaheadad adder.

Overall, the Dadda multiplier compromises speed and hardware complexity, making it a popular choice for performing high-performance multiplication operations in DSP applications where speed and area efficiency are two of the most important criteria.

## BRENT-KUNG ADDER

The Brent-Kung adder utilizes A parallel prefix structure, which is a high-speed carry-look ahead adder design. This architecture helps to decrease the critical path latency within the system. It is frequently utilized in digital signal processing (DSP) applications in which high-speed arithmetic operations are necessary. The Brent-Kung adder can perform its operations more quickly than typical ripple-carry adders by producing carry signals in parallel using a tree-based structure. This allows for the addition of multi-bit operands to be performed more quickly. We will represent two binary integers, A and B, that will be joined together. Each of these numbers has a total of N bits. Numerous steps make up the Brent-Kung adder design. Each stage produces intermediate carry signals, which are then utilized to calculate the final total.

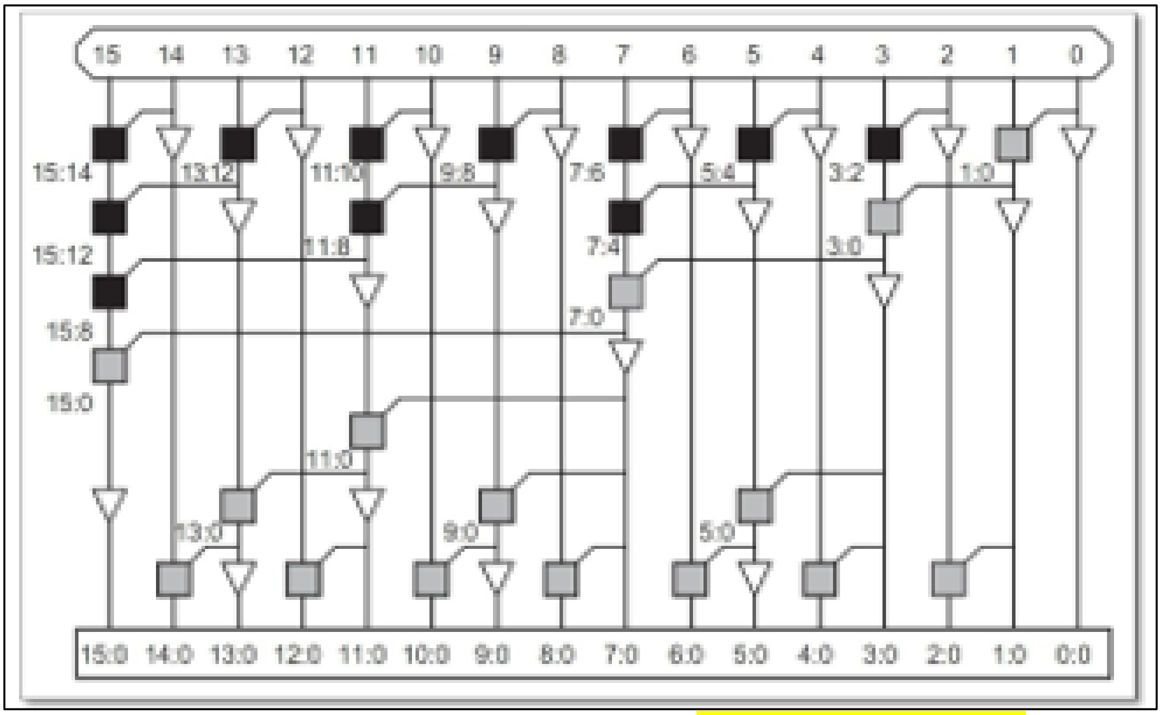

***Figure 3:*** *Block diagram of Brent Kung Adder. Please provide clear image*

Fig. 3 shows the block diagram of Brent Kung Adder and describes the operation of the Brent-Kung adder by using the equations that are included below:

- Generate (G) and Propagate (P) signals:

$G_i = A_i \cdot B_i$ (Generate)

$P_i = A_i + B_i$ (Propagate)

$G_i$ and $P_i$ represent the generated and propagated signals for the *i*th bit position.

- Generate the Carry Signals:

$C_{i,j} = G_i + P_i \cdot C_{i-1,j}$ for $i = 1, 2, \ldots, N$ and $j = 1, 2, \ldots, \lfloor \log_2(N) \rfloor$

$C_{i,j}$ represents the carry signal generated at stage j for the ith bit position.

- Calculate the sum:

$Si = Ai \oplus Bi \oplus Ci - 1, \lfloor \log2(N) \rfloor$ for $i= 1, 2, . . . ,N$

Where $S_i$ represents the $i$th bit of the sum.

In these equations, the logical AND operation is represented by the symbol ·, the logical OR operation is represented by the symbol +, and the symbol represents the logical XOR operation ⊕. The carry signals are created in parallel throughout the various stages of the adder, enabling the computation of the final total to be carried out effectively without the requirement for carry propagation over a ripple-carry chain.

By adopting a parallel prefix structure, the Brent-Kung adder can decrease the critical path latency and enable high-speed addition of multi-bit operands. As a result, it is well suited for digital signal processing applications, which are characterized by the importance of performing quick arithmetic operations for real-time signal processing.

## METHODOLOGY

The Dadda multiplier, which is well-known for the area efficiency it achieves due to its parallel design, is utilized in the suggested system to make the most of its efficiency. By employing this multiplier architecture, the system intends to maximize available resource consumption and preserve its computations' correctness. To complement this strategy, a Brent-Kung adder is incorporated into the last stages of merging. This adder offers significant benefits regarding the speed at which it processes information and the amount of power it consumes.

The implementation of the system is carried out with the help of Verilog HDL, a hardware description language frequently used in digital design projects. This approach makes it possible to integrate with Xilinx tools smoothly, making it easier to do detailed simulations and analyses of the system's performance. The system's efficiency is evaluated by comparing its performance indicators, such as space power consumption and processing speed, to those of a standard booth-encoded multiplier. This comparison is carried out using a rigorous assessment method.

A robust solution for high-speed and resource-efficient digital signal processing applications is the goal of the system that has been developed. This is accomplished by combining a Dadda multiplier's strengths with a Brent-Kung adder's efficiency. The potential of the system to outperform conventional ways is investigated via careful assessment and analysis, which prepares the way for further developments in the design and implementation of digital signal processing (DSP) systems.

## RESULTS

Xilinx ISE 14.7 was a comprehensive design suite for FPGA design and implementation, offering tools for design entry, synthesis, implementation, verification, and programming. Fig. 4 shows the Simulation result of the FIR filter designed using the Xilinx ISE tool.

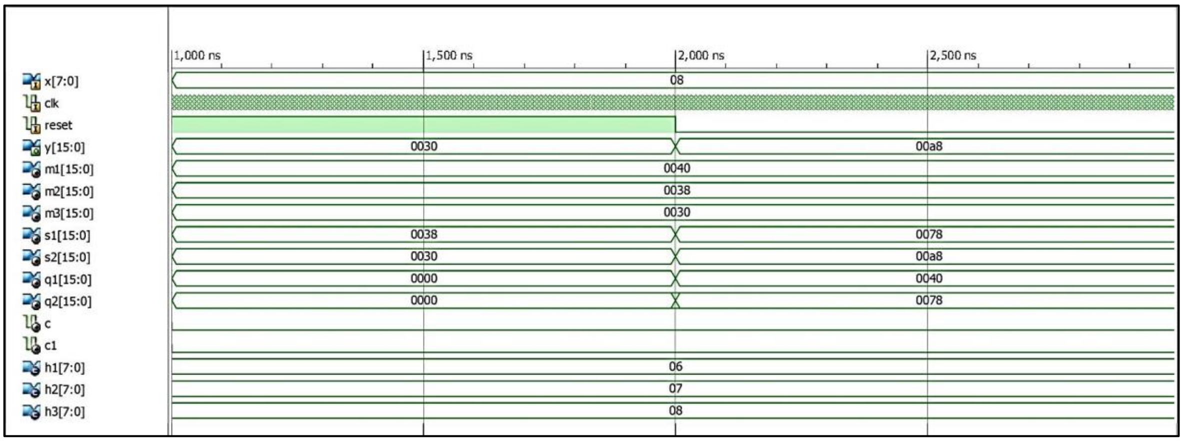

***Figure 4:*** *Simulation result of FIR filter.*

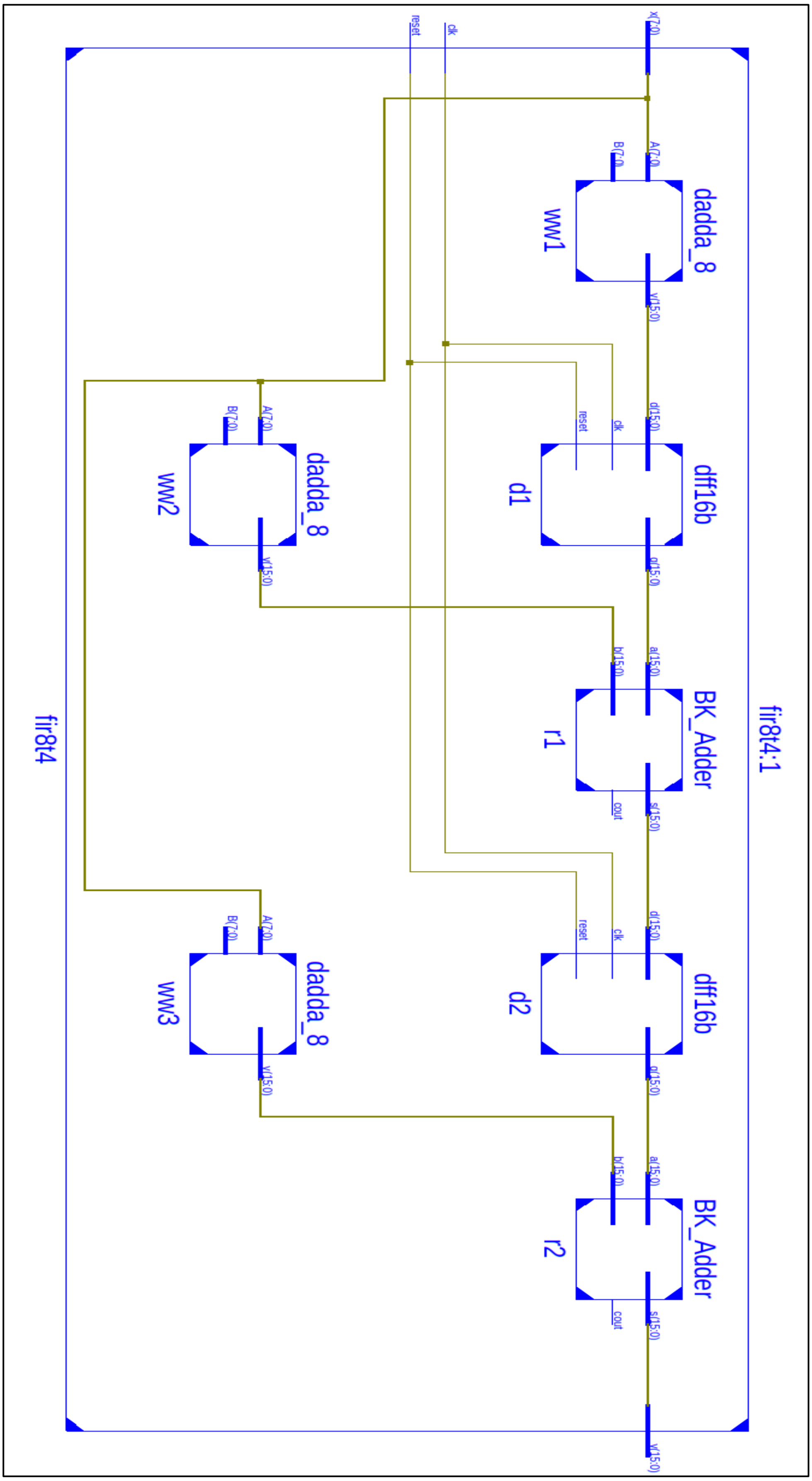

**Figure 5:** *RTL schematic of FIR filter.*

Fig. 5 shows the RTL schematic of the FIR filter, illustrates that it takes an 8-bit input signal `x` and processes it through a series of multiplication and addition operations with predefined coefficients `h0`, `h1`, `h2`, and `h3`. These coefficients generate partial products in four Dadda Multiplier modules (`ww1`, `ww2`, `ww3`, `ww4`). The partial products are then accumulated in three BK_Adder modules (`r1`, `r2`, and `r3`) to produce intermediate sum results (`s1`, `s2`, `s3`). Before further processing, these intermediate results are registered using D Flip-Flops (`d1`, `d2`, `d3`). Finally, the output signal `y' is assigned the last intermediate sum result `s3` value, representing the filtered output. The RTL schematic depicts data flow through multipliers, adders, and registers to perform the FIR filtering operation on the input signal.

***Table 1:*** *Resource utilization. Provide table citation*

| Resource Parameters | Existing FIR Filter [1] | Proposed FIR Filter |
|---|---|---|
| No. of Slices (out of 4656) | 180 used (4%) | 47 used (1%) |
| No. of 4 input LUTs (out of 9312) | 273 used (3%) | 87 used (0%) |
| No. of bonded IOBs (out of 232) | 171 | 26 used (11%) |
| POWER(W) | 21.3 | 11.13 |

The Dadda multiplier, well-known for its higher area efficiency due to its parallel design, is incorporated into the suggested system. A Brent Kung adder is also used to increase speed and reduce power consumption during the final merging phases. The system, implemented in Verilog HDL, provides seamless digital design and simulation by utilizing Xilinx utilities. The performance study comprehensively analyses critical variables such as space usage, power consumption, and processing speed. This is done by comparing the suggested technique to a conventional Booth-encoded multiplier.

A durable solution for high-performance digital signal processing applications is presented by the system that has been described. This solution is achieved by combining the area efficiency of a Dadda multiplier with the speed and power benefits of a Brent-Kung adder. Utilizing Verilog HDL for implementation and Xilinx tools for simulation, the system is subjected to a full assessment compared to conventional techniques. This examination shows the system's potential to transcend existing efficiency and computing capabilities methodologies.

## CONCLUSION

In conclusion, the implementation strategy that has been given for FIR filters, which incorporates a Dadda multiplier and a Brent-Kung adder, provides a potential answer to the issues provided by traditional multiplier and adder designs. The technique that has been described is capable of satisfying the rigorous requirements of modern signal processing and communication systems. This is accomplished by maximizing silicon area usage and boosting computational efficiency. In the future, research may investigate the possibility of further refining and optimizing the approach described to broaden its applicability to a broader variety of digital signal processing applications. In the end, the incorporation of contemporary multiplier and adder designs paves the way for the development of digital filtering systems that are both more effective and more powerful.

The designed FIR filter demonstrates remarkable enhancements across various parameters compared to the existing filter. Regarding slice utilization, the existing filter consumes 180 slices out of 4656, constituting 4% of available resources, whereas the proposed filter utilizes only 47 slices, representing a mere 1% utilization. Regarding 4-input Look-Up Table (LUT) utilization, the existing filter utilizes 273 LUTs out of 9312, equivalent to 3%, while the proposed filter uses 87 LUTs, amounting to 0% utilization. When considering bonded Input/Output Blocks (IOBs), the existing filter utilizes 171 out of 232, indicating a high utilization rate of 70%, whereas the proposed filter utilizes only 26, representing an 11% utilization. Regarding power consumption, the existing FIR filter consumes 21.3 watts, while the proposed filter demonstrates significantly lower power consumption at 11.13 watts.

Overall, the proposed FIR filter shows a substantial reduction in resource utilization and power consumption compared to the existing filter, suggesting it is a more efficient option for implementation.

## REFERENCES

1. Hegde, R., & Shanbhag, N. R. (2004). A voltage overscaled low-power digital filter IC. *IEEE Journal of Solid-State Circuits*, *39*(2), 388-391. https://doi.org/10.1109/JSSC.2003.821775
2. Shim, B., Sridhara, S. R., & Shanbhag, N. R. (2004). Reliable low-power digital signal processing via reduced precision redundancy. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, *12*(5), 497-510. https://doi.org/10.1109/TVLSI.2004.826201
3. Whatmough, P. N., Das, S., Bull, D. M., & Darwazeh, I. (2012). Circuit-level timing error tolerance for low-power DSP filters and transforms. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, *21*(6), 989-999. https://doi.org/10.1109/TVLSI.2012.2202930
4. Nanduri, S., & Kamaraju, M. (2023). Efficient VLSI implementation of modular neural network based hybrid median filter. *Soft Computing*, 1-15. https://doi.org/10.1007/s00500-023-08762-0
5. Rao, K. A., Pandit, M., & Purohit, N. (2022, December). Efficient 3-parallel polyphase odd length FIR filter using Brent Kung adder and Booth multiplier for VLSI applications. In *2022 IEEE 9th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)* (pp. 1-5). IEEE. https://doi.org/10.1109/UPCON56432.2022.9986374
6. Bonetti, A., Teman, A., Flatresse, P., & Burg, A. (2017). Multipliers-driven perturbation of coefficients for low-power operation in reconfigurable FIR filters. *IEEE Transactions on Circuits and Systems I: Regular Papers*, *64*(9), 2388-2400. https://doi.org/10.1109/TCSI.2017.2698138
7. Yadeeswaran, K. S., Prakalya, D., Mithun Mithra, N., Athukuri, C., & Mohan, N. (2023, March). FIR and IIR Filter Design Using Modified Dadda Multiplier. In *Proceedings of Fourth International Conference on Communication, Computing and Electronics Systems: ICCCES 2022* (pp. 185-195). Singapore: Springer Nature Singapore. https://doi.org/10.1007/978-981-19-7753-4_15
8. Pushpalatha, P., & Babulu, K. (2024, February). Design and Implementation of Adders and Multipliers for DSP Applications. In *International Conference on Communications and Cyber Physical Engineering 2018* (pp. 53-69). Singapore: Springer Nature Singapore. https://doi.org/10.1007/978-981-99-7137-4_6
9. Sambamurthy, N., & Kamaraju, M. (2023). FPGA implementation of high performance image de-noising filter. *Analog Integrated Circuits and Signal Processing*, 1-12. https://doi.org/10.1007/s10470-023-02208-1

**CITE THIS ARTICLE**

M. Deepika Krishna, et al. (2024). Efficient Implementation of FIR Filter Utilizing Dadda Multiplier and Brent Kung Adder, *Journal of Controller and Converters*, 9(2), 8-16.