ENERGY EFFICIENT VLSI DESIGN OF PARALLEL FIR FILTERS USING ENHANCED BOOTH MULTIPLIER AND ADAPTIVE PREFIX ADDERS FOR REAL-TIME DSP SYSTEMS

[1]B.Pooja, [2]Dr. K. Sravan Abhilash

[1]M.Tech Scholar, Department of Electronics and Communication Engineering, CMR Engineering College, Medchal, Kandlakoya, Seethariguda, Telangana, India

[2]Associate Professor, Department of Electronics and Communication Engineering, CMR Engineering College, Medchal, Kandlakoya, Seethariguda, Telangana, India

ABSTRACT: Finite impulse response (FIR) filters are extensively used in various digital signal processing applications such as digital audio, image processing, data transmission, biomedical etc. In some applications, the FIR filter circuit must be capable to operate at high sample rates, while in other applications, the FIR filter circuit must be a low power circuit operating at moderate sample rates. An energy-efficient parallel FIR filter architecture designed for high-speed VLSI (Very Large Scale Integration) applications is presented in this system. The 3-parallel Fast FIR Algorithm (FFA) polyphase structure with optimized arithmetic units are integrated in this proposed design. The reduction of number of partial products and signed multiplication efficiency enhancement is carried out by multiplication by using an Enhanced Booth Multiplier. The Wallace Tree structure is used to compress Partial products performs Modified Brent-Kung Adder, offering a balance of high speeds and low area overhead while accumulation. In different application requirements, modular design allows programmable parallelism (1–4) enables scalable performance. Therefore, the proposed design achieves higher throughput with reduced power consumption when compared with conventional array multiplier and ripple-carry adder based FIR filters in real-time DSP and low-power VLSI implementations.

KEYWORDS: **Finite Impulse Response (FIR), Ripple-Carry Adder, Programmable Parallelism, Booth Multiplier, Fast FIR Algorithm (FFA), VLSI, Wallace Tree Structure**

I. INTRODUCTION

During digital signal processing, a digital filter is essential. The most often used kind of digital filter implementation in software is the finite impulse response (FIR) filter[1]. Thus, parallel FIR filters are employed to adjust the sample rate or power consumption according to the needs of the application. Throughput may be increased and power consumption reduced using digital parallel FIR filters. Researchers have been concentrating on the FIR filter over the last several decades. Although there has been a lot of research on parallel FIR filters, the most of it focuses on using the fast FIR method to minimize the number of multipliers[2]. Considering the intention of increasing complexity By iterating tiny filtering structures and lowering the number of computational (i.e., adder and multiplier) units through lowering the number of associated parallel subfilter units, the conventional technique thus achieves fast FIR algorithms (FFA)[3]. Additionally, revised FFA were created for the use of linear phase parallel FIR filters. Specifically, these algorithms, which were suggested for odd-length FIR filters, used the idea of symmetric coefficients, which resulted in a halving of the multipliers across the subfilter units and an increase in the total amount of adders for the pre/post processing blocks.

Two of the more crucial parts of the filter design comprise the multiplier and adder[4]. The technical complexity associated with parallel polyphase FIR filter construction may be reduced, according to recent studies. The filter's overall computing speed and power consumption are determined by the performance provided by the multiplier and adder blocks. Since the effectiveness of the FIR filter depends significantly on the operation of multipliers and adders, several kinds of adders and multipliers can be found in digital circuits[5]. This allows for a technology-independent method to designing an area-efficient parallel polyphase FIR filter and DSP applications. Digital filters were a crucial component of DSP. Actually, one of the main factors contributing to DSP's rise in popularity is their exceptional performance[6]. Signal restoration and signal separation are the two applications for filters. Whenever a signal is tainted by noise, interference, or other signals, signal separation must be performed. Consider, for instance, a gadget that measures the electrical activity of a developing baby's heart (EKG) although it is still within the womb. It is probable that the mother's breathing and pulse will taint the raw signal[7]. These signals have to be separated using a filter in order to be examined separately. Whenever a signal has been altered in any manner, signal restoration is utilized. For instance, to better capture the sound as it really happened, an audio recording having inadequate specifications could be filtered. Another instance might be the first appearance of a picture taken with a shaky camera or an incorrectly focused lens. Analog or digital filters may be used to combat these issues[8].

Analog filters are inexpensive, quick, and have a wide dynamic range in terms of frequency as well as amplitude. In contrast, the degree of performance that may be attained with digital filters is significantly higher[9]. Compared to analog filters, digital filters may perform thousands of times better. This has a significant impact on how filtering difficulties are handled. When using analog filters, the focus is on managing electronic constraints such resistor and capacitor precision and stability. Comparatively speaking, digital filters have become so effective that their performance is usually disregarded. The focus switches to the signals' limits and the theoretical problems with processing them. In many digital signal processing (DSP) methods, multiplying a parameter by a set of previously determined constant coefficients constitutes a standard operation[10].

Multiplication can be the most costly operation for DSP algorithms when compared to more frequent operations like addition, subtraction, employing delay elements, etc. The quantity of silicon within the integrated circuit, or the amount of logic resources needed, is traded off with the computation's speed[11]. Assuming the same number of logic resources, multiplication takes longer than most other operations. It also takes longer when each operation has to be finished in the identical period of time. When performing multiplication using two arbitrary variables, a generic multiplier is required[12]. Nevertheless, we can use the characteristics of binary multiplication to construct a less costly logic circuit that is functionally equal to only asserting the constant upon one input of a general multiplier while multiplying with a known constant. Multiplication can be quite costly, therefore in many circumstances, employing a less expensive solution just only multiplication still yields considerable savings when taking into account the full logic circuit. Additionally, based on the application, multiplication may be the dominating operation. High-performance digital signal processing (DSP) circuits are becoming more and more in demand within the field of Very Large Scale Integration (VLSI) applications[13].

Finite Impulse Response (FIR) filters are an essential part of many digital signal processing systems because of its stability, linear phase response, and simplicity of use[14]. However, new methods for creating effective and optimized FIR filters are desperately needed as the demands for increased throughput and reduced power usage increase. Finite Impulse Response (FIR) filter may greatly increase processing speed and lower power consumption during digital signal processing. Multiple filter outputs may be computed simultaneously thanks to the above structure, which splits the filtering operation into three parallel branches. Efficiency was further maximized by including a Booth multiplier plus a BrentKung adder into this design[15]. Overall Brent-Kung adder lowers the total addition latency during the filtering process because of its logarithmic delay and low wiring requirements. During the meanwhile, this multiplier speeds up the necessary multiplications for FIR filtering by effectively handling signed integers and minimizing the amount of partial products. In the area of Very Large Scale Integration (VLSI) applications, the development of effective digital filters plays a significant role in signal processing activities. The Finite Impulse Response (FIR) filter constitutes a significant filter type that is frequently employed for data compression, signal equalization, and noise reduction. These parts are designed to maximize the filter's computational speed, which makes it ideal for VLSI circuit real-time processing applications.

II. LITERATURE SURVEY

R. Sharma, S. Yadav and S. K. Saha, et al. [16] presents Honey Badger Algorithm (HBA) to find the optimized coefficients of FIR linear phase filters, i.e., Band Pass Filter (BPF), Band Stop Filter (BSF), Low Pass Filter (LPF), and High Pass Filter (HPF). With its quick convergence and reduced number of tuning parameters, HBA is a promising new metaheuristic method. HBA, which imitates the foraging behavior of the honey badger is likely to maintain the balance between the two phrases of exploration and exploitation. HBA computes the optimal coefficients of the specified filter using the digging and honey-searching modes. As a consequence, the provided work is able to accomplish the ideal frequency performance characteristics with the maximum stop band attenuation and the negligible pass band ripple. In other words, the HBA designed filter can perform better than filters using other methods because it has fewer ripples in passband and narrower transition width, which enhances its frequency response.

B. S. M. Ali, Z. K. Farej and A. M. Ibrahim, et al. [17] use of MATLAB Filter Design and Analysis Tool (FDAT) to obtain the desired filter coefficients, then using the open-source Arduino Due microcontroller to implement the FIR filter function in software. The filter coefficients, which are obtained from MATLAB FDAT are truncated to be accepted by Arduino software as a single line of software. Using the ADC and the DAC available in Arduino, the FIR filter analog input and output can be both observed and analyzed by the student to verify the designed filter characteristics. A real time FIR filter (with an order of 10) was realized in Arduino software using signal frequencies less than 5KHz.

A. Anand, S. Yadav and S. K. Saha, et al.[18] a new heuristic optimization algorithm called Artificial Hummingbird Optimization (AHA) is proposed for the design of FIR low and high pass filters(LPF and HPF). The AHA optimization technique models the different flight skills and foraging strategies of hummingbird. A set of optimized filter coefficients is generated using the AHA algorithm in order to achieve the desired specifications. The obtained results of the AHA algorithm are compared with PSO, BFO, GWO and BAT algorithms in terms of passband ripples(PBR), stopband attenuation(SBA), stopband ripples (SBR) and transition width (TW).

G. Capizzi and G. L. Sciuto, et al. [19] presents a novel design method for multidimensional finite-impulse response

(FlR) filters based on the representation of the desired frequency response by using a Gabor system generated by a Gaussian function. A direct synthesis of the desired filter frequency response is obtained from standard design specifications. Further, we report some comparative tests with the well known Parks-McClellan's methodology in order to illustrate the concrete advantages of the novel design approach that we have proposed. With such an approach, the design procedure turns out to be especially simple to implement, as it employs closed design formulas which directly link filter coefficients to design specifications, and moreover, affords a better pass-band flatness than any known design approach. It also yields stop-band attenuations and grants control over critical frequencies, which turn out being very close to those obtainable with Parks-McClellan method. The enhanced performance of the 2-D filters thus obtained are well suited for space applications.

R. Durgagopal and D. N. Rao, et al. [20] design and implementation of a power-optimized low pass digital FIR filter based on the ripple carry adder and the radix 4 modified booth algorithm are presented in this work, as well as use of the Five Modular Redundancy (FMR) approach. This paper aims to apply the parallel adding method (PAM) is used and the radix 4 modified booth algorithm on low pass digital FIR filter and Five Modular Redundancy (FMR) method is used to correct the error in transmitted signals and compare the existing low pass digital FIR filter. These filtering utilised error correcting codes with the forward error - correcting ability, resulting in enhanced chip space and fault - tolerant. The recommended methodology employs less redundancy module than previous techniques and decreases the size of such filtering by around 21.18 percent, decreases the cost of systems development by nearly the same proportion. Our method is written in Verilog, and it is implemented on a Vivado Basys 3 board from Xilinx. The Xilinx vivado 2021.1 tool is used to calculate the following parameters: area, power, delay, and LUT. This paper discusses the synthesis as well as the outcomes.

S. Rooban, M. Nagesh, M. V. S. L. Prasanna, K. Rayudu and G. D. Sai, et al. [21] description of the multiplication of two binary numbers of size 128-bits each using Radix-4 Booth's Algorithm is presented in this paper. Booth Encoder circuit, Partial Product Generator tree, and Carry save adders are the main building blocks of any Booth Multiplier. Carry Save Adder is useful for adding all the partial products that are obtained. The radix-4 Booth algorithm is used to decrease the partial products which are generated intermediately. Higher representation radix results in smaller amounts of digits for a number to be represented. Radix-4 algorithm of booth helps to reduce the partial products when compared to Radix-2 booth's algorithm by a factor of half. Radix- 4 booth's algorithm is recommended because most of the parameters performances such as delay, area and power are more efficient for the radix-4 algorithm when compared to the higher radices (like radix-8's booth's algorithm or radix-16 booth's algorithm).

R. Orugu, S. Padamata, Y. Kollati, L. Nakka, Y. Nunna and R. Sai Midhilesh Mamidi, et al.[22] proposes an approximate radix-8 Booth multiplier designed to enhance performance and reduce complexity compared to traditional counterparts. Specifically tailored for signed-unsigned numbers, the proposed multiplier undergoes evaluation via simulation using Xilinx Vivado software with Artix-7 FPGA Board. Leveraging techniques such as modified Booth encoding and parallel partial product generation, it aims to achieve reduced delay and power consumption. The application of this multiplier is demonstrated through integration into Sobel edge detection. Quantitative metrics, including speedup factor and area reduction, are provided to assess performance improvements over existing systems. Furthermore, the proposed design is highly accurate, detecting 98.87% of edges, demonstrating its practicality and usefulness in digital image processing applications.

P. Sanjana, M. Ramesh, A. Kale, A. A. Anita and P. Sasipriya, et al. [23] 8-bit approximate booth multipliers have been proposed using i) Approximate radix-4 Booth Encoders (ABE), ii) truncated partial products and iii) recursive Carry Look Ahead (CLA) adder for final addition to generate the final product terms. The accuracy of the multiplier is scaled by introducing approximation factor in terms of truncation and utilization of ABEs. State of Art accurate booth multiplier is also implemented for comparison purpose. Error performance of the proposed multipliers has been evaluated by means of Normalized Mean Error Distance (NMED). To show the feasibility of employing approximate circuits for real time applications, image multiplication using approximate booth multipliers have been realized. The performance of each of the multipliers is demonstrated by measuring PSNR and SSIM and compared using an image processing application.

B. V. Mahesh and T. Srivasarao, et al.[24] Booth multipliers' fundamental design, this is tested on ASIC-based platforms. FPGA-based hardware accelerators cannot give the required performance gain. This can be achieved by using FPGAs and ASICs. For approximation 6-input Look Up Table (LUT) and carry chains of the FPGAs are used. In the existing method, it is possible to have data with 40% of error probability reduction and is not acceptable in logic design for signal processing or for data communications. To overcome this error in end product, instead of Booth multiplier, AHL (Adaptive Hold Logic) Booth multiplier is recommended. With the AHL booth multiplier, the error Probability is expected to reduce to a great extent. Means that100% of accurate data reception is possible at the output compared to existing approximate Booth multiplier. AHL Booth multiplier improves the operation speed and reduces the delay.

G. Park, J. Kung and Y. Lee, et al.[25] present a novel design methodology of cost-effective approximate radix-4 Booth multipliers, which can significantly reduce the power consumption of error-resilient signal processing tasks. In contrast that the prior studies only focus on the approximation of either the partial product generation with encoders or the partial product reductions with compressors, the proposed method considers two major processing steps jointly by forcing the generated error directions to be opposite to each other. As the internal errors are naturally balanced to have zero mean, as a result, the proposed approximate Booth multiplier can minimize the required processing energy under the same number of approximate bits compared to the previous designs. Simulation results on FIR filtering and image classification applications reveal that the proposed approximate Booth multiplier shows the most attractive energy-performance trade-offs, achieving 28% and 34% of energy reduction compared to the exact Booth multiplier, respectively, with negligible accuracy loss.

III. FRAMEWORK OF ENERGY EFFICIENT VLSI DESIGN OF PARALLEL FIR FILTERS USING ENHANCED BOOTH MULTIPLIER AND ADAPTIVE PREFIX ADDERS FOR REAL-TIME DSP SYSTEMS

In this section, framework of energy efficient VLSI design of parallel FIR Filters using enhanced booth multiplier and adaptive prefix adders for real-time

DSP systems is observed in figure 1. FIR Top chooses parallelism (1–4) and supplies coefficients. It manages sample buffering and timing. FIR Core instantiates N parallel MAC datapaths (configurable) and handles polyphase mapping using FFA. The Per-MAC multiplier performs signed multiplication using Booth encoding and produces partial products in Booth Multiplier. In final summation, Wallace / PPG reduces partial products quickly (carry-save adders/Wallace tree) to two rows. In the final, the adder/accumulator uses a spanning-tree prefix adder for fast, area-efficient in Brent–Kung Adder. The parallel outputs back are merged into the correct time-sequence; optional FIFO or downsampler in Output.

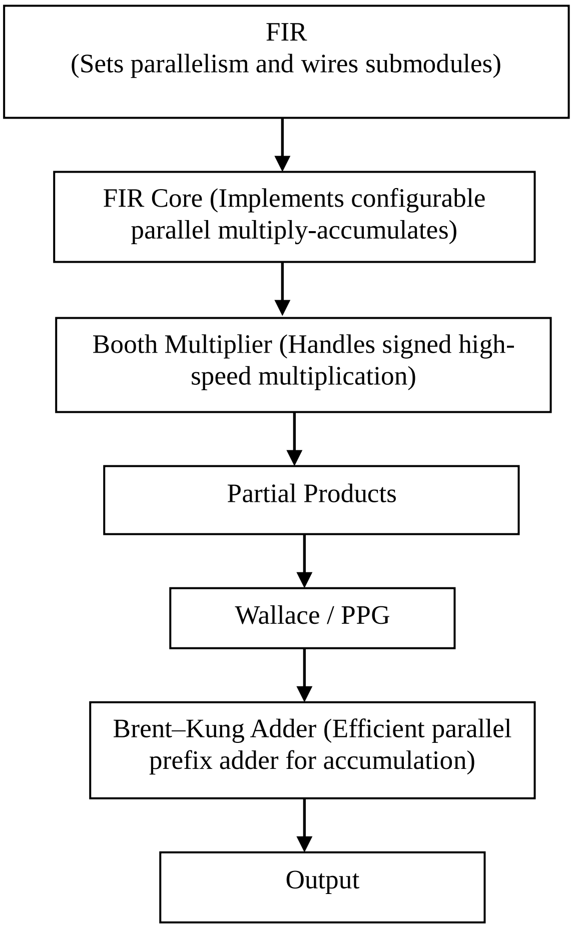
FIR
(Sets parallelism and wires submodules)
FIR Core (Implements configurable parallel multiply-accumulates)
Booth Multiplier (Handles signed high-speed multiplication)
Partial Products
Wallace / PPG
Brent–Kung Adder (Efficient parallel prefix adder for accumulation)
Output

Figure.1: Framework of Energy Efficient VLSI Design Of Parallel FIR Filters Using Enhanced Booth Multiplier And Adaptive Prefix Adders For Real-Time DSP Systems

In VLSI, FIR stands for Finite Impulse Response. It refers to a type of digital filter commonly used in signal processing, known for its stability and ease of implementation in hardware. FIR filters are characterized by their finite-duration impulse response, meaning their output eventually settles to zero after a finite time when subjected to an impulse input. FIR filters are inherently stable because their output is a finite sum of weighted past inputs, guaranteeing no feedback loops that could cause instability.FIR filters can be implemented using basic arithmetic operations (multiplication and addition) and delay elements (flip-flops), making them suitable for VLSI design. VLSI design focuses on optimizing the filter's performance (speed, power consumption, area) by using techniques like pipelining, coefficient symmetry, and specialized multiplier architectures (e.g., Booth multiplier, Wallace tree multiplier).

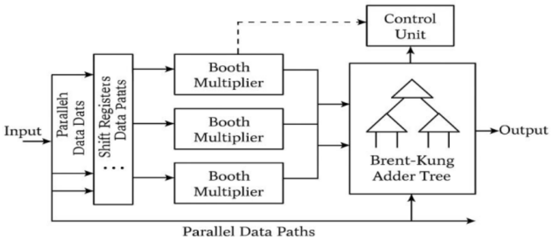

Figure. 2: Proposed FIR Filter Architecture with Programmable Parallelism

Booth's multiplication algorithm is an efficient method for multiplying binary numbers, particularly useful for handling both positive and negative numbers in 2's complement representation. It optimizes the multiplication process by reducing the number of additions and subtractions needed compared to standard multiplication algorithms. This is achieved by encoding the multiplier in a way that identifies patterns of consecutive 1s and 0s. Booth's algorithm is designed to work with signed binary numbers represented in 2's complement, where negative numbers are represented by inverting all bits of the positive number and adding 1. After each bit pair check and the corresponding operation (addition or subtraction), the registers holding the partial product and the multiplier are shifted one bit to the right.

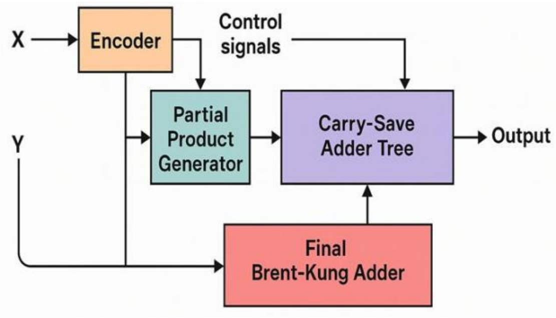

Figure.3: Enhanced Booth Multiplier Block Diagram with All Internal Components

The partial products are intermediate results generated during the multiplication of two binary numbers. These products are then summed using specialized adder circuits to produce the final multiplication result. This process is crucial for efficient hardware implementation of multipliers in digital circuits. After reduction, the remaining partial products are added using a fast carry-propagation adder, such as a carry-lookahead adder, to produce the final multiplication result. A Brent–Kung adder is a parallel adder made in a regular layout with an aim of minimizing the chip area and ease of manufacturing. The addition of n-bit number can be performed in time with a chip size of area. thus making it a good-choice adder with constraints on area and maximizing the performance. The Brent-Kung adder employs a parallel prefix structure, which means it computes carry signals concurrently for multiple bits, unlike ripple-carry adders where carry signals propagate sequentially. The adder's carry computation network is designed with a logarithmic depth, meaning the number of stages (and thus the delay) increases logarithmically with the input size. Compared to some other parallel prefix adders like Kogge-Stone, the Brent-Kung adder aims to minimize wiring complexity by strategically overlapping computation stages.

IV. RESULT ANALYSIS

In this section, result analysis of energy efficient VLSI design of parallel FIR Filters using enhanced booth multiplier and adaptive prefix adders for real-time DSP systems is observed.

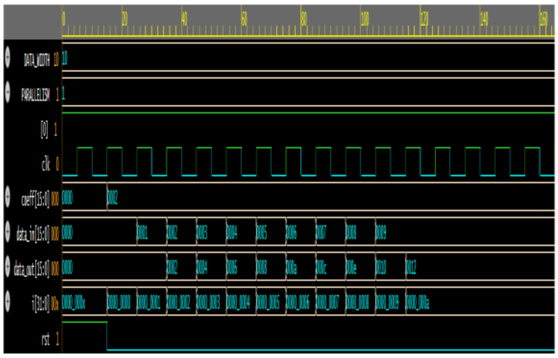

Figure.4: Output wave form for parallelism -1

The waveform proves that the FIR filter is working with input numbers are fed one by one, multiplied by coefficient 0x02, and the results appear at data_out. The output matches expected multiplication results (2, 4, 6, 8, …) and it resets properly, with correct clocking and the filter datapath behaves as designed.

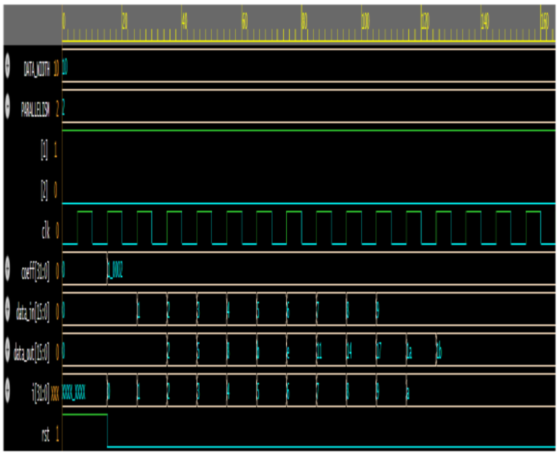

Figure.5: Output wave form for parallelism -2

The waveform shows the FIR filter operating with parallelism-2. The two input samples are processed at a time and reset correctly, outputs are generated in sync with the clock and confirming proper parallel FIR operation.

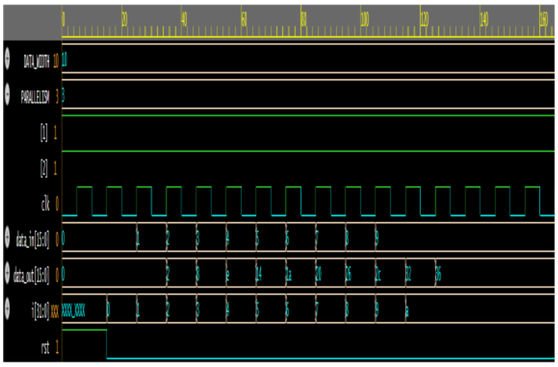
0
20
40
60
80
100
120
140
160
DATA_WIDTH 10 10
PARALLELISM 3 3
[1] 1
[2] 1
clk 0
data_in[15:0] 0 0 1 2 3 4 5 6 7 8 9
data_out[15:0] 0 0 2 8 e 14 1a 20 26 2c 32 36
i[31:0] XXX XXXX_XXXX 0 1 2 3 4 5 6 7 8 9 a
rst 1

Figure.6: Output wave form for parallelism -3

The waveform shows the FIR filter running with parallelism -3. It uses three input samples are processed in parallel. The outputs appear faster when compared to parallelism -1 or 2. It shows that higher parallelism improves throughput while maintaining correct filter operation.

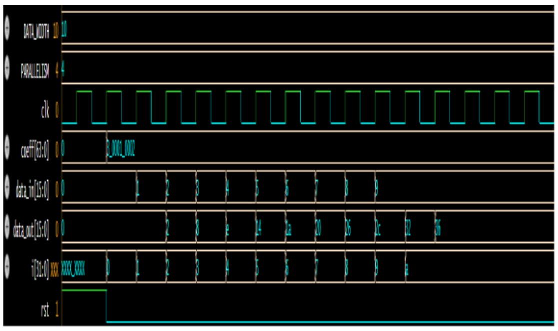

Figure.7: Output wave form for parallelism -4

The waveform shows the FIR filter working with parallelism -4. It shows maximum parallelism achieves the highest throughput, producing results every clock cycle after reset.

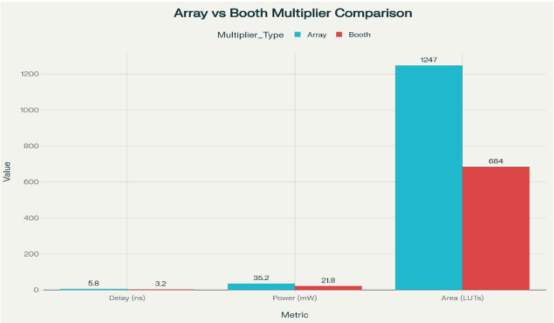

Figure.8: Performance Comparison: Array vs Booth Multipliers

Array multipliers have higher delay and area, while Booth multipliers reduce partial products, power, and time. Hence, Booth multipliers are better for DSP applications.

V. CONCLUSION

In this section, energy efficient VLSI design of parallel FIR Filters using enhanced booth multiplier and adaptive prefix adders for real-time DSP systems is concluded. The parallel FIR filter architecture optimized for energy efficiency and high performance is demonstrated in this design. An Enhanced Booth Multiplier and a Modified Brent-Kung Adder with polyphase FFA-based parallel structure is integrated in this proposed design. This proposed design achieves reduced critical path delay and lower power consumption when compared with traditional FIR implementations. The Booth encoding is used to minimize the number of partial products during Wallace tree and prefix adder enhances computational speed and efficiency. Therefore, this proposed approach provides a high throughput and speed by FIR filter solution.

VI. REFERENCES

[1] G. D. Cain, A. Yardim and F. J. Harris, "Live Demonstration: Comparison of Three FIR Digital Filter Sharpening Techniques," *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, Seville, Spain, 2020, pp. 1-1, doi: 10.1109/ISCAS45731.2020.9180956.

[2] Q. Zhang, P. Qu, S. Sun, E. Yu and W. Yan, "Design of Full Digital Signal Generator and FIR Digital Filter System Based on FPGA," *2022 41st Chinese Control Conference (CCC)*, Hefei, China, 2022, pp. 4501-4506, doi: 10.23919/CCC55666.2022.9901637.

[3] C. Klöck, "FIR Filter Design Method Based on LASSO to Adjust the Number of Non-zero Coefficients," *2021 International Conference on Computer, Information and Telecommunication Systems (CITS)*, Istanbul, Turkey, 2021,

pp. 1-5, doi: 10.1109/CITS52676.2021.9618507.

[4] N. Biswas, S. Dhabal and P. Venkateswaran, "Analysis of Area Efficient Parallel FIR Filters using FPGA," *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, Delhi, India, 2023, pp. 1-5, doi: 10.1109/ICCCNT56998.2023.10308163.

[5] B. P. Stošić, "The Effects of Coefficient Quantization on New CIC FIR Filters," *2020 55th International Scientific Conference on Information, Communication and Energy Systems and Technologies (ICEST)*, Niš, Serbia, 2020, pp. 151-154, doi: 10.1109/ICEST49890.2020.9232824.

[6] M. Hu, X. Jiang and Z. Li, "A Resource-efficient FIR Filter Design Based on an RAG Improved Algorithm," *2023 5th International Conference on Circuits and Systems (ICCS)*, Huzhou, China, 2023, pp. 209-212, doi: 10.1109/ICCS59502.2023.10367312.

[7] Y. Li, T. Jin, Y. Zhao and X. Ma, "Design of FIR Filter Based on FPGA," *2023 5th International Conference on Robotics, Intelligent Control and Artificial Intelligence (RICAI)*, Hangzhou, China, 2023, pp. 167-170, doi: 10.1109/RICAI60863.2023.10489185.

[8] M. Lemke, T. Dörlemann, R. Nowak, J. Aigner and S. Frei, "Comparison of FIR Filter Lengths and Adaptation Methods in Digital Active EMI Filters based on Delay-Compensated Digital Gate Control Signals," *2024 International Symposium on Electromagnetic Compatibility – EMC Europe*, Brugge, Belgium, 2024, pp. 1100-1105, doi: 10.1109/EMCEurope59828.2024.10722456.

[9] P. S. Apostolov, A. K. Stefanov and M. S. Bagasheva, "Efficient FIR Filter Synthesis Using Sigmoidal Function," *2019 X National Conference with International Participation (ELECTRONICA)*, Sofia, Bulgaria, 2019, pp. 1-4, doi: 10.1109/ELECTRONICA.2019.8825598.

[10] G. Wang, "Implementation of FIR Digital Filter Algorithm in C Language on DSP Platform," *2024 5th International Conference on Information Science, Parallel and Distributed Systems (ISPDS)*, Guangzhou, China, 2024, pp. 446-450, doi: 10.1109/ISPDS62779.2024.10667488.

[11] S. Yadav, M. Kumar, R. Yadav and A. Kumar, "Efficient Design of Sparse FIR Digital Filter," *2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, Greater Noida, India, 2020, pp. 647-651, doi: 10.1109/ICACCCN51052.2020.9362902.

[12] P. Zahradnik, "Analytical Design of Two-Dimensional Maximally Flat Notch FIR Filters," *2019 6th International Conference on Signal Processing and Integrated Networks (SPIN)*, Noida, India, 2019, pp. 116-119, doi: 10.1109/SPIN.2019.8711699.

[13] Y. S. Mudhafar and S. H. Abdulnabi, "High Performance FIR and IIR Filters Based on FPGA for 16 Hz Signal Processing," *2023 5th International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, Istanbul, Turkiye, 2023, pp. 1-6, doi: 10.1109/HORA58378.2023.10156676.

[14] T. C. Singh and M. Kumar, "Digital FIR Filter Designs," *2021 Asian Conference on Innovation in Technology (ASIANCON)*, PUNE, India, 2021, pp. 1-5, doi:10.1109/ASIANCON51346.2021.9544766.

[15] G. Li *et al.*, "Data Rearrangement Optimized FIR High-Pass Filter Implemented on FPGA," *2024 8th International Conference on Communication and Information Systems (ICCIS)*, Shenzhen, China, 2024, pp. 6-10, doi: 10.1109/ICCIS63642.2024.10779421.

[16] R. Sharma, S. Yadav and S. K. Saha, "Optimal FIR Filter Design using Honey Badger Optimization Algorithm," *2023 6th*

International Conference on Information Systems and Computer Networks (ISCON), Mathura, India, 2023, pp. 1-6, doi: 10.1109/ISCON57294.2023.10112079.

[17] B. S. M. Ali, Z. K. Farej and A. M. Ibrahim, "Implementing FIR Filters using Arduino Due Platform for Educational Purposes," *2019 2nd International Conference on Engineering Technology and its Applications (IICETA)*, Al-Najef, Iraq, 2019, pp. 49-54, doi: 10.1109/IICETA47481.2019.9012990.

[18] A. Anand, S. Yadav and S. K. Saha, "An approach for Linear Phase FIR low Pass and High Pass Filter Design using AHA Algorithm," *2022 6th International Conference On Computing, Communication, Control And Automation (ICCUBEA*, Pune, India, 2022, pp. 1-5, doi: 10.1109/ICCUBEA54992.2022.10010872.

[19] G. Capizzi and G. L. Sciuto, "A Novel 2-D FIR Filter Design Methodology Based on a Gaussian-Based Approximation," in *IEEE Signal Processing Letters*, vol. 26, no. 2, pp. 362-366, Feb. 2019, doi: 10.1109/LSP.2019.2891965

[20] R. Durgagopal and D. N. Rao, "Power Optimized Low Pass Digital FIR Filter Using Five Modular Redundancy and Parallel Adding Methods," *2022 International Conference on Intelligent Controller and Computing for Smart Power (ICICCSP)*, Hyderabad, India, 2022, pp. 1-6, doi: 10.1109/ICICCSP53532.2022.9862465.

[21] S. Rooban, M. Nagesh, M. V. S. L. Prasanna, K. Rayudu and G. D. Sai, "Implementation of 128-bit Radix-4 Booth Multiplier," *2021 International Conference on Computer Communication and Informatics (ICCCI)*, Coimbatore, India, 2021, pp. 1-7, doi: 10.1109/ICCCI50826.2021.9457004.

[22] R. Orugu, S. Padamata, Y. Kollati, L. Nakka, Y. Nunna and R. Sai Midhilesh Mamidi, "FPGA Design and Implementation of Approximate Radix-8 Booth Multiplier," *2024 Third International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE)*, Ballari, India, 2024, pp. 01-06, doi: 10.1109/ICDCECE60827.2024.10548931.

[23] P. Sanjana, M. Ramesh, A. Kale, A. A. Anita and P. Sasipriya, "Design and Evaluation of Error Tolerant Booth Multipliers for Image Processing Applications," *2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT)*, Tirunelveli, India, 2022, pp. 1414-1418, doi: 10.1109/ICSSIT53264.2022.9716363.

[24] B. V. Mahesh and T. Srivasarao, "Performance Evaluation of FFT through Adaptive Hold Logic (AHL) Booth Multiplier," *2023 International Conference for Advancement in Technology (ICONAT)*, Goa, India, 2023, pp. 1-6, doi: 10.1109/ICONAT57137.2023.10080290.

[25] G. Park, J. Kung and Y. Lee, "Simplified Compressor and Encoder Designs for Low-Cost Approximate Radix-4 Booth Multiplier," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 70, no. 3, pp. 1154-1158, March 2023, doi: 10.1109/TCSII.2022.3217696.