Design & Simulation of Power Efficient FIR Filter With Shift-Add Based RTL Modeling and Optimization

Mrs. B. Swapna[1], P. Sai Praneeth[2], P. Ujwal[3], P. Manidhar Reddy[4], P. Akhila Reddy[5]
[1]*Assistant Professor, Dept. of ECE, TKR College of Engineering and Technology*
[2,3,4,5]*Student, Dept. of ECE, TKR College of Engineering and Technology*

***Abstract*—This paper presents the design and simulation of a power efficient Finite Impulse Response (FIR) filter using a shift-add based Register Transfer Level (RTL) modeling optimization. Conventional FIR filters rely on multiplier-based architectures, which result in high power consumption and increased hardware complexity. To overcome these limitations, the proposed design replaces multipliers with shift and add operations, thereby reducing area and improving power efficiency. The FIR filter is modeled using Verilog HDL and implemented using FPGA design tools such as Xilinx Vivado.**

***Index Terms*—FIR Filter, Shift-Add Technique, RTL Modeling, Low Power Design, FPGA Implementation.**

I. INTRODUCTION

Digital Signal Processing (DSP) has become an essential component in modern electronic systems, enabling efficient analysis, modification, and synthesis of signals in applications such as wireless communication, multimedia systems, biomedical instrumentation, and radar processing. Among the various DSP building blocks, digital filters play a vital role in tasks such as noise reduction, signal shaping, and bandwidth limitation. One of the most widely used classes of digital filters is the Finite Impulse Response (FIR) filter, primarily due to its inherent stability, absence of feedback, and capability to achieve an exact linear phase response. An FIR filter produces an output that depends solely on the present and past input samples, which can be mathematically represented as a convolution between the input sequence and a finite set of filter coefficients. This non recursive nature ensures that FIR filters are always stable and free from issues such as limit cycle oscillations. However, the implementation of FIR filters typically involves a large number of multiplication and accumulation operations, particularly in high order filters. In hardware implementations, these multipliers contribute significantly to increased power consumption, silicon area, and computational delay. With the rapid advancement of portable and battery-operated devices, including mobile communication systems and Internet of Things (IoT) applications, power efficiency has emerged as a critical design constraint.

II. METHODOLOGY

The methodology adopted in this work focuses on the design and implementation of a power efficient Finite Impulse Response (FIR) filter using a shift add based Register Transfer Level (RTL) modeling approach. The design process begins with the mathematical formulation of the FIR filter, where the output is expressed as the convolution of the input signal with a finite set of coefficients. In conventional FIR filter implementations, this operation requires multiple multiplication and accumulation stages, which significantly increase hardware complexity and power consumption, particularly in high order filters. To address this limitation, the proposed methodology replaces the conventional multiplier-based architecture with a shift add based realization. In this approach, constant coefficient multiplications are represented using a combination of binary shift and addition operations. Since shifting corresponds to simple bit manipulation and does not require complex hardware resources, and addition operations are less resource intensive compared to multipliers, the overall computational complexity is significantly reduced. This transformation enables the realization of a multiplier less FIR filter, which is highly suitable for low power and resource constrained applications. The architectural implementation of the

proposed FIR filter is carried out at the Register Transfer Level using Verilog Hardware Description Language. The design consists of a series of delay elements forming a tapped delay line, which stores the present and past input samples. These delayed samples are then processed through shift operations corresponding to the coefficient values, followed by accumulation using adder units to generate the final output.

The data flow within the system is synchronized with a clock signal, ensuring proper sequencing of operations and reliable hardware implementation. The overall architecture of the proposed system, including the FIR filter structure and shift-add module, is illustrated in Fig. 1 and Fig. 2, respectively. The block diagram of the FIR filter depicts the flow of input signals through delay elements, coefficient multiplication units implemented using shift-add operations, and the accumulation stage. Similarly, the shift-add module demonstrates the internal structure used to replace conventional multipliers with efficient shift and addition logic. These architectural representations provide a clear understanding of the data flow and hardware optimization employed in the proposed design.

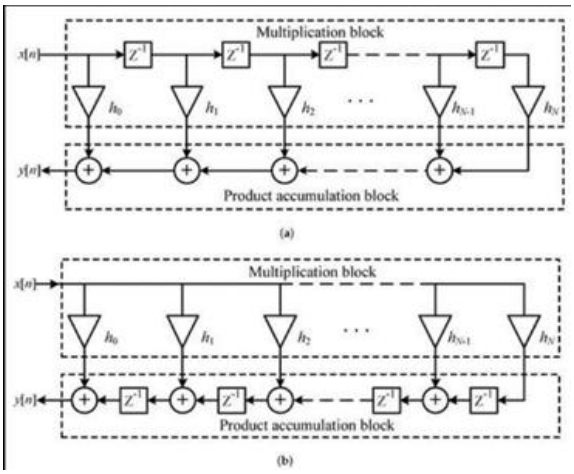

Figure1: Shift Add Implementation

III. IMPLEMENTED DESIGN

The implementation of the proposed power-efficient Finite Impulse Response (FIR) filter is carried out using a shift-add based Register Transfer Level (RTL) modeling approach. The design is described using Verilog Hardware Description Language and synthesized using FPGA design tools. The implemented architecture replaces conventional multiplier units with shift and adds operations, thereby significantly reducing hardware complexity and power consumption. The entire design is structured to ensure efficient data flow, synchronization with clock signals, and optimal utilization of hardware resources.

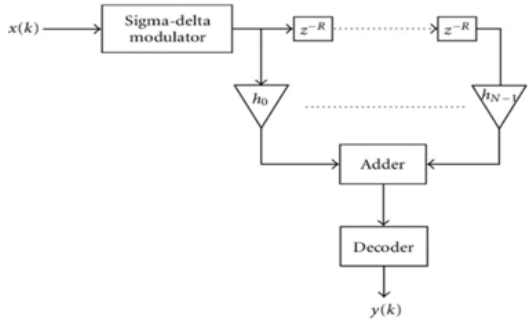

Figure:2: FIR Block Diagram

The FIR filter architecture consists of a series of delay elements forming a tapped delay line, which stores the present and past input samples. These delayed samples are processed sequentially through shift operations corresponding to the coefficient values. Each coefficient is represented in a form that enables multiplication through bit shifting and addition, eliminating the need for dedicated multipliers. The shifted outputs are then accumulated using adder units to produce the final filtered output. The overall structure ensures that the output is generated as a weighted sum of input samples, consistent with the FIR filter formulation.

The RTL implementation is designed using synchronous logic, where all operations are triggered on the rising edge of the clock signal. A set of registers is used to store intermediate values, ensuring proper timing and data propagation through the system. The shift-add operations are implemented using combinational logic, while the delay elements are realized using flip-flops. This separation of sequential and combinational components enhances the modularity and scalability of the design. The implemented FIR filter is simulated using FPGA-based tools to verify functional correctness. Test input sequences are applied, and the corresponding output responses are analyzed to ensure accurate filtering operation.

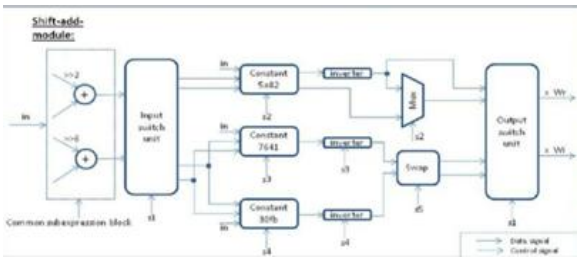

Figure:3: Shift-Add Module

The synthesis results provide detailed information regarding resource utilization, timing performance, and power consumption. The proposed design demonstrates improved efficiency in terms of reduced power usage and hardware area when compared to conventional multiplier-based FIR filter implementations. These techniques reduce redundant arithmetic operations and minimize the number of adders required in the design.

IV. RESULT&DISCUSSION

The proposed power-efficient FIR filter based on shift-add RTL modeling is implemented and verified using FPGA design tools. Functional simulation is carried out by applying a sequence of input samples, and the corresponding output responses are observed to validate the correctness of the filter operation. The simulation results confirm that the designed FIR filter produces accurate outputs corresponding to the given input sequence, thereby verifying the functional integrity of the system. The waveform analysis illustrates the relationship between input and filtered output signals, demonstrating the effectiveness of the implemented architecture in performing the desired filtering operation.

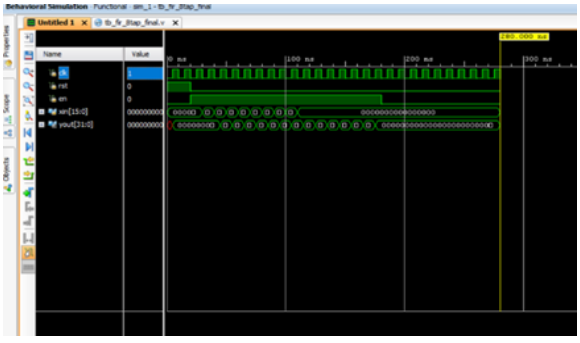

Figure:4: Simulation Output Waveform of FIR Filter

The implementation results are analyzed in terms of power consumption, area utilization, and timing performance. The synthesis reports indicate that the proposed shift-add based FIR filter significantly reduces hardware resource utilization by eliminating multipliers and replacing them with simpler shift and add operations. As observed from the project results, the power consumption of the shift-add based design is considerably lower compared to the conventional multiplier-based FIR filter implementation. This reduction is primarily attributed to the decreased switching activity and simplified arithmetic operations involved in the design.

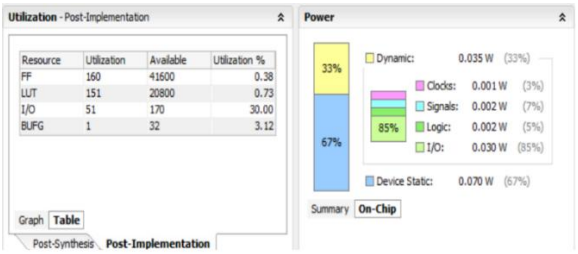

Figure:5: Power Consumption of FIR Filter

The area utilization is also reduced due to the absence of dedicated multiplier units, which are typically resource-intensive components in digital hardware. The use of shift registers, adders, and combinational logic ensures efficient utilization of FPGA resources such as lookup tables (LUTs) and flip-flops. In addition to power and area improvements, the proposed design also demonstrates satisfactory timing performance, as the simplified operations contribute to reduced propagation delay and improved processing speed. A comparative analysis between the conventional multiplier-based FIR filter and the proposed shift-add based FIR filter clearly highlights the advantages of the proposed approach. The multiplier-based design offers flexibility in coefficient representation but suffers from higher power consumption and increased hardware complexity. In contrast, the shift-add based design achieves significant improvements in power efficiency and area reduction, making it more suitable for low-power and real-time applications.

V.CONCLUSION

In this paper, a power-efficient Finite Impulse Response (FIR) filter has been designed and implemented using a shift-add based Register Transfer Level (RTL) modeling approach. The proposed design focuses on reducing the hardware complexity and power consumption associated with conventional multiplier-based FIR filter architectures. By replacing multipliers with shift and add operations, the design effectively minimizes the use of resource-intensive components, resulting in improved area efficiency and reduced switching activity.

The implementation of the FIR filter using Verilog HDL and FPGA design tools demonstrates that the proposed architecture achieves accurate filtering performance while significantly lowering power consumption. The simulation and synthesis results validate the functional correctness of the design and highlight its advantages in terms of reduced hardware utilization and enhanced computational efficiency. The use of optimized data flow, delay elements, and efficient coefficient representation further contributes to the overall performance improvement of the system.
A comparative analysis with the conventional multiplier-based FIR filter confirms that the proposed shift-add based design offers superior performance in terms of power and area, while maintaining acceptable speed and accuracy. These improvements make the design highly suitable for low-power and real-time digital signal processing applications, particularly in portable and embedded systems.

The use of RTL modeling ensures that the design is modular, scalable, and easily adaptable to different system requirements. The architecture can be extended to higher-order filters and integrated into complex digital signal processing systems without significant modifications. This flexibility, combined with improved efficiency, makes the proposed design a practical solution for modern DSP applications. In conclusion, the shift-add based RTL implementation of the FIR filter offers a reliable and efficient alternative to conventional designs. The achieved improvements in power consumption, area utilization, and computational efficiency validate the effectiveness of the proposed methodology. Hence, the developed system can be effectively utilized in applications such as communication systems, biomedical signal processing, and embedded platforms, where low power operation and optimized hardware performance are essential.

REFERENCES

[1] V. Oppenheim and R. W. Schafer, Digital Signal Processing. Englewood Cliffs, NJ, USA: Prentice Hall, 1975.

[2] J. G. Proakis and D. G. Manolakis, Digital Signal Processing: Principles, Algorithms, and Applications, 4th ed. Upper Saddle River, NJ, USA: Pearson Education, 2007.

[3] S. K. Mitra, Digital Signal Processing: A Computer-Based Approach. New York, NY, USA: McGraw-Hill, 2006.

[4] K. K. Parhi, VLSI Digital Signal Processing Systems: Design and Implementation. New York, NY, USA: Wiley-Interscience, 1999.

[5] Xilinx Inc., Vivado Design Suite User Guide, 2020.

[6] N. J. Fliege, Multirate Digital Signal Processing. New York, NY, USA: Wiley, 1994.

[7] M. Meher, "Low-power FIR filter architectures using distributed arithmetic," IEEE Transactions on Circuits and Systems, vol. 60, no. 2, pp. 337–345, 2013.

[8] R. Hartley, "Subexpression sharing in FIR filter implementation," IEEE Transactions on Circuits and Systems II, vol. 43, no. 10, pp. 677–688, 1996.

[9] T. Saramäki, "Finite impulse response filter design," in Handbook for Digital Signal Processing. New York, NY, USA: Wiley, 1993.

[10] "FPGA-based FIR filter design using shift-add technique," International Journal of Engineering Research and Technology (IJERT), 2018.

[11] L. R. Rabiner and B. Gold, Theory and Application of Digital Signal Processing. Englewood Cliffs, NJ, USA: Prentice Hall, 1975.

[12] Venkataramani and S. R. S. Prasad, "Multiplierless FIR filter design using canonical signed digit (CSD) representation," IEEE Transactions on Circuits and Systems II, vol. 45, no. 10, pp. 1325–1329, 1998.

[13] Y. Voronenko and M. Püschel, "Multiplierless multiple constant multiplication," ACM Transactions on Algorithms, vol. 3, no. 2, pp. 1–19, 2007.

[14] H. Samueli, “An improved search algorithm for the design of multiplier less FIR filters,” IEEE Transactions on Circuits and Systems, vol. 36, no. 7, pp. 1044–1047, 1989.
[15] R. Meher, “Efficient FPGA implementation of FIR filters using shift and add technique,” IEEE Transactions on Very Large-Scale Integration (VLSI) Systems, vol. 21, no. 3, pp. 385–390, 2013.
[16] S. Haykin, Adaptive Filter Theory, 5th ed. Upper Saddle River, NJ, USA: Pearson, 2013.
[17] P. P. Vaidyanathan, Multirate Systems and Filter Banks. Englewood Cliffs, NJ, USA: Prentice Hall, 1993.
[18] Xilinx Inc., 7 Series FPGA Overview, Datasheet, 2018.
[19] M. C. Johnson, “Design of low-power digital filters using pipelining and parallel processing,” IEEE Signal Processing Letters, vol. 8, no. 5, pp. 135–138, 2001.
[20] K. Roy and S. Prasad, Low-Power CMOS VLSI Circuit Design. New York, NY, USA: Wiley, 2000.