ISSN 2319-5991 www.ijerst.com

Peer Reviewed, Referred & Indexed Journal

Research Paper

LOW-POWER FIR FILTER USING APPROXIMATION TECHNIQUES FOR ENERGY-EFFICIENT VLSI SYSTEMS

[1]G. RAMESH, [2]Dr.V. PRASANTH

[1]PG Scholar, [2]Associate Professor, Department of Electronics & Communication Engineering, PRAGATI ENGINEERING COLLEGE, Surampalem, E.G. District, A.P.- 533437, India

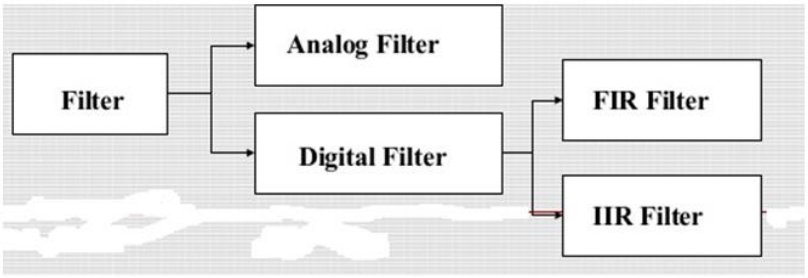
ABSTRACT

The increasing demand for portable electronics, Internet of Things (IoT) devices, wearable systems, communication equipment, and embedded applications has created a strong requirement for high-performance digital circuits with reduced power consumption and hardware complexity. Finite Impulse Response (FIR) filters are fundamental digital signal-processing blocks widely employed in communication, audio processing, image processing, biomedical systems, and data-processing applications. However, conventional FIR filters involve repeated multiplication and accumulation operations, resulting in considerable power dissipation, area utilization, and computational delay. Therefore, efficient arithmetic architectures are essential for energy-constrained Very Large Scale Integration (VLSI) systems. This work presents a low-power FIR filter based on approximation techniques applied to arithmetic operations. The proposed architecture incorporates an array multiplier and approximate arithmetic units to reduce unnecessary hardware activity while maintaining acceptable computational accuracy. High-speed adder architectures, particularly parallel-prefix and Ladner–Fischer structures, are considered to improve addition speed and reduce carry-propagation delay. The design is described using Verilog HDL and implemented and evaluated using the Xilinx Vivado environment. Functional simulation, RTL analysis, synthesis, timing evaluation, and power analysis are performed to assess the effectiveness of the proposed architecture. The simulation demonstrates correct functional operation of the approximate FIR filter while indicating improvements in hardware efficiency and delay. The approximation introduces a controlled computational error, but this error can be tolerated in error-resilient DSP applications. The proposed architecture therefore provides a practical trade-off among power, area, speed, and accuracy, making it suitable for energy-efficient VLSI implementations.

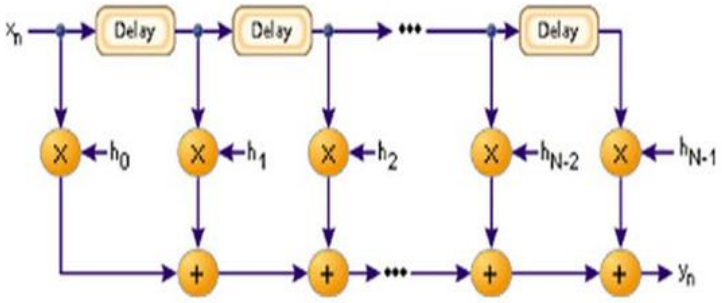
Keywords: FIR filter, approximate computing, low-power VLSI, approximate adder, array multiplier, Ladner–Fischer adder, parallel-prefix adder, Verilog HDL, Xilinx Vivado, energy-efficient DSP.

I. INTRODUCTION

Digital filters are fundamental components of signal-processing systems because they selectively suppress unwanted signal components while preserving required information. Filters are widely used in telecommunications, audio systems, radar, control systems, image processing, and other electronic applications. Among digital filters, Finite Impulse Response (FIR) filters are particularly important because they provide inherent stability, feed-forward implementation, and the possibility of achieving linear-phase characteristics through appropriate coefficient symmetry [1]. FIR filters are extensively employed in discrete-time signal-processing operations, including filtering, communication systems, transforms, and error-processing applications [2]. Classical FIR filter design methods enable designers to obtain low-pass, high-pass, band-pass, and band-stop responses according to specified frequency requirements [3]. However, the implementation of an FIR filter requires multiple constant-coefficient multiplications followed by accumulation operations. Consequently, the multiplier and adder structures significantly influence the overall area, power consumption, and speed of the resulting VLSI architecture [4]. Wallace introduced an efficient multiplication structure based on parallel reduction of partial products, establishing an important foundation for high-speed arithmetic hardware [5]. Booth-based multiplication techniques subsequently provided efficient approaches for signed multiplication and reduced-area arithmetic structures [6]. Constant multiplication can also be optimized through shift-and-add decomposition, thereby reducing the number of general-purpose multiplication operations [7]. Such optimization is especially useful for FIR filters because their coefficients remain constant during normal operation.



Classification of Filters



Block diagram of FIR filter

The continuing scaling of VLSI technology has increased the importance of power, area, and timing optimization. Conventional exact arithmetic guarantees numerical correctness, but exact computation is not always necessary for applications such as image, audio, speech, and other error-resilient DSP systems [8]. Approximate computing exploits this characteristic by intentionally relaxing arithmetic accuracy to obtain improvements in power, delay, and hardware complexity [9]. Approximate arithmetic circuits can therefore provide useful energy-performance trade-offs for computationally intensive systems [10]. In FIR filters, approximation can be introduced into multipliers, adders, partial-product generation, or accumulation structures. The uploaded project specifically investigates approximate arithmetic together with efficient multiplier and adder architectures, implemented using Verilog HDL and evaluated through Xilinx Vivado. The proposed work focuses on reducing unnecessary switching and arithmetic complexity while preserving acceptable filtering behavior. The architecture further employs high-speed prefix-based addition, with the Ladner–Fischer structure selected for high-performance arithmetic operation [11]. Thus, the overall objective is to develop an FIR filter that provides a balanced combination of low power consumption, reduced hardware complexity, improved speed, and acceptable output accuracy.

II. LITERATURE SURVEY

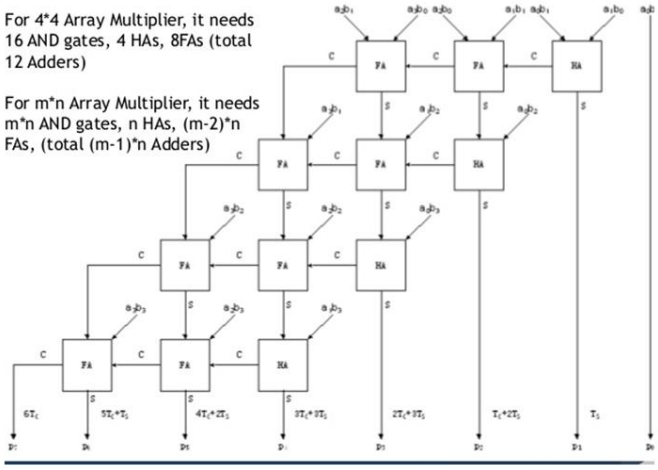
Early FIR-filter research concentrated on systematic filter design and efficient implementation. McClellan, Parks, and Rabiner developed a computer-based method for designing optimum linear-phase FIR filters with different frequency specifications, establishing a systematic foundation for practical FIR design [12]. Wanhammar presented important concepts related to DSP integrated circuits and their implementation requirements [13]. Wallace proposed a fast multiplier based on parallel generation and reduction of partial products, demonstrating the importance of arithmetic architecture in high-speed digital systems [14]. Gallagher and Swartzlander investigated high-radix Booth multipliers and reduced-area adder trees, addressing the area and speed limitations associated with multiplication [15]. Nguyen and Chatterjee introduced number-splitting combined with shift-and-add decomposition to optimize power and hardware in linear DSP synthesis [16]. Their approach demonstrated that reducing the number of arithmetic operations can significantly improve implementation efficiency. Hartley investigated subexpression sharing using canonical signed-digit multipliers and showed that common subexpressions in filter structures can substantially reduce the number of arithmetic operators [17]. Park and Kang further investigated minimal signed-digit representations for digital-filter synthesis, demonstrating the potential for reducing constant-multiplication complexity [18]. Dempster and Macleod studied minimum-adder multiplier blocks specifically for FIR filters, emphasizing the importance of minimizing arithmetic resources [19]. Voronenko and Püschel subsequently investigated multiplierless multiple-constant multiplication as an efficient approach for constant-coefficient arithmetic [20].

More recent research has increasingly focused on approximate arithmetic and low-power implementation. Aksoy et al. investigated exact and approximate algorithms for optimizing area and delay in multiple-constant multiplication [21]. Aksoy, Güneş, and Flores further explored exact and approximate search algorithms for multiple-constant multiplication [22]. Ladner and Fischer introduced a parallel-prefix computation structure that forms the theoretical basis of the Ladner–Fischer adder used in high-speed arithmetic [23]. Parallel-prefix architectures reduce carry

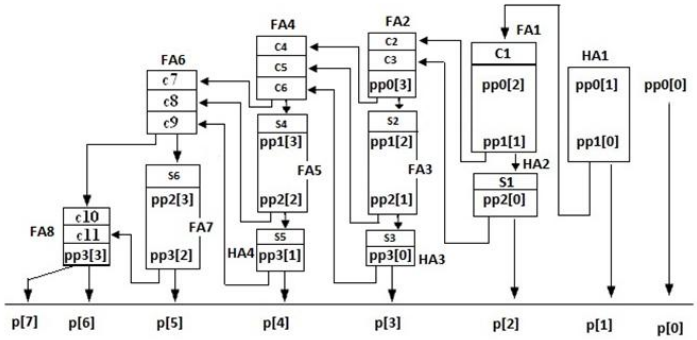
computation depth compared with conventional ripple-based structures, making them suitable for high-speed VLSI arithmetic. The uploaded project discusses parallel-prefix structures and specifically selects the Ladner–Fischer architecture for high-performance addition. Erdogan and Arslan investigated low-power implementation techniques for FIR filtering structures and reported power reductions through switching-activity minimization [24]. Low-power FIR implementations using Booth multipliers, shift/add structures, folding, and low-power adders have also demonstrated the importance of arithmetic-unit selection in reducing dynamic power [25]. Truncated multiplier techniques have been explored to reduce FIR implementation cost while maintaining acceptable frequency response and output accuracy [26]. Approximate multipliers have subsequently been investigated for DSP applications, including FIR filters, with reported power savings at controlled accuracy degradation [27]. Recent surveys classify approximate adders, multipliers, dividers, and other arithmetic blocks according to their error and energy characteristics [28], while broader evaluations emphasize that approximate arithmetic can improve energy efficiency in error-resilient applications [29]. These studies motivate the present work, which combines approximation with efficient multiplier and high-speed adder architectures to develop an energy-efficient FIR filter.

II. PROPOSED SYSTEM

The proposed system is an approximate FIR filter architecture designed to reduce power consumption, hardware complexity, and arithmetic delay while maintaining acceptable filtering accuracy. The conventional FIR structure consists primarily of delay elements, coefficient multipliers, and an accumulation stage. In the proposed architecture, input samples are passed through delay elements implemented using flip-flops, while constant coefficients are processed using multiplier structures. The uploaded project identifies the array multiplier as the basic multiplier architecture, where partial products are generated using AND operations and subsequently accumulated using adder structures. Instead of performing all arithmetic operations with complete precision, approximation is introduced into selected arithmetic units. The approximate half-adder replaces the conventional XOR-based sum operation with an OR-based approximation, while approximate full-adder logic similarly simplifies the sum computation. This reduces logic complexity and switching activity, thereby providing potential improvements in power and area. The approximation level can be selected according to the error tolerance of the intended DSP application.

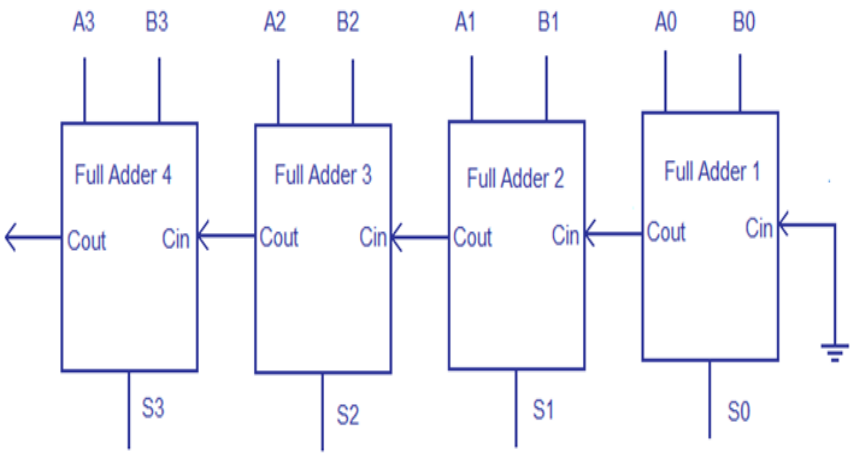


Block diagram of array multiplier

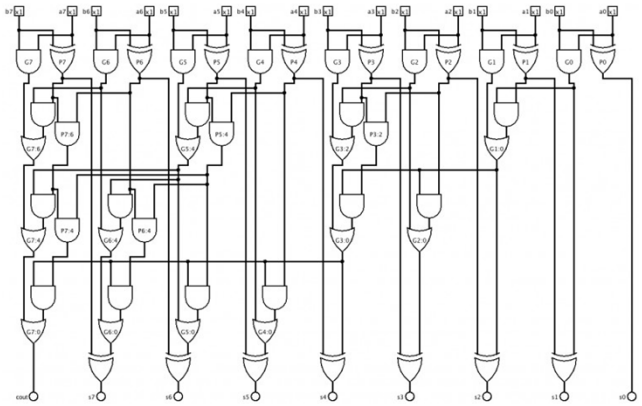


Bit array multiplier

The second major component of the proposed architecture is the high-speed addition stage. Conventional ripple-carry addition suffers from carry propagation delay because each bit depends on the carry generated by the preceding bit. The project therefore investigates carry-lookahead and parallel-prefix architectures as alternatives. Parallel-prefix structures calculate generate and propagate information through tree-based structures, reducing the effective carry-computation depth. Among these structures, the Ladner–Fischer adder is selected as the proposed high-speed adder. The Ladner–Fischer architecture uses black and gray cells arranged in a prefix-tree structure to generate carries efficiently. The complete FIR architecture is described using Verilog HDL, functionally simulated, synthesized, and evaluated using Xilinx Vivado. The design flow includes HDL coding, simulation, synthesis, RTL analysis, timing analysis, and power estimation. This implementation methodology allows the proposed approximate FIR filter to be compared with conventional arithmetic structures in terms of functional correctness, hardware utilization, power, and timing.



Ripple-Carry Adder

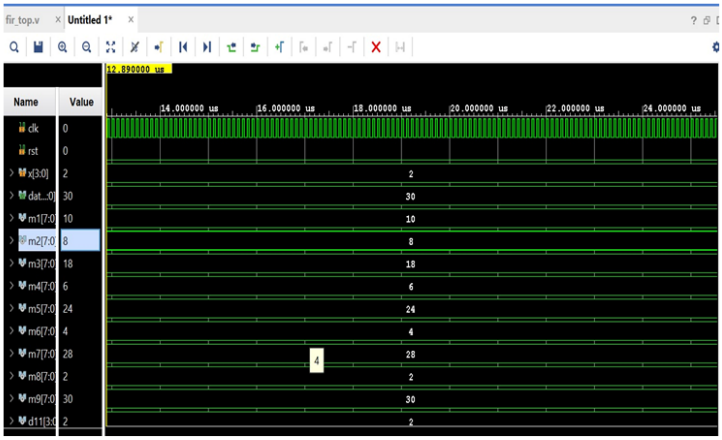


Lardner fischer adder VLSI

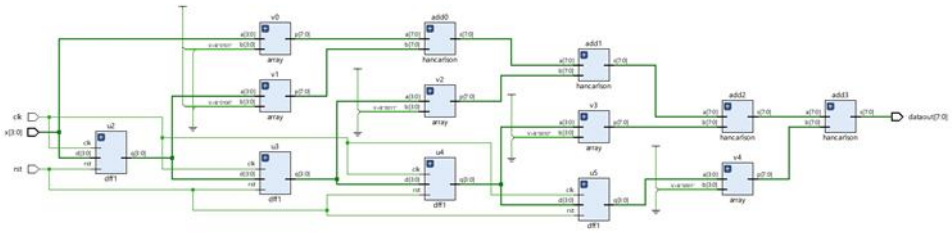
IV. SIMULATION RESULTS & ANALYSIS

The proposed approximate FIR filter was functionally simulated and synthesized using the Xilinx Vivado environment. The project report includes the simulation waveform, RTL schematic, power report, setup-time analysis, and hold-time analysis as the principal evaluation results. The simulation confirms the functional operation of the approximate FIR filter, while the RTL representation verifies the synthesized structural organization of the arithmetic units. The report specifically records a power value of **7.998 W** for the stated implementation configuration; however, the surrounding report does not provide sufficient numerical information to independently calculate a percentage power reduction against the existing design. Timing analysis considers propagation, setup, and hold behavior. The results indicate that the proposed architecture provides reduced delay with a comparatively simpler implementation, consistent with the objective of using efficient arithmetic structures. Approximation inevitably introduces a small numerical error, but the project considers this acceptable for applications in which energy efficiency and hardware performance are more important than exact arithmetic precision. The resulting trade-

off demonstrates the suitability of the architecture for DSP and image-processing applications where controlled computational error can be tolerated.



Simulation Results of Approximate FIR Filter



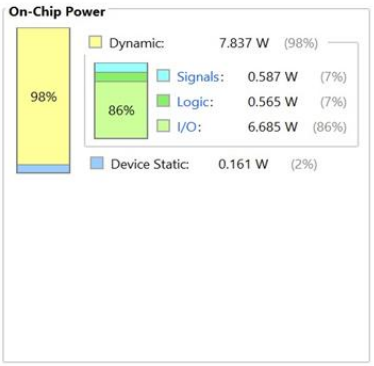
RTL Schematic

EXISTING POWER REPORT

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

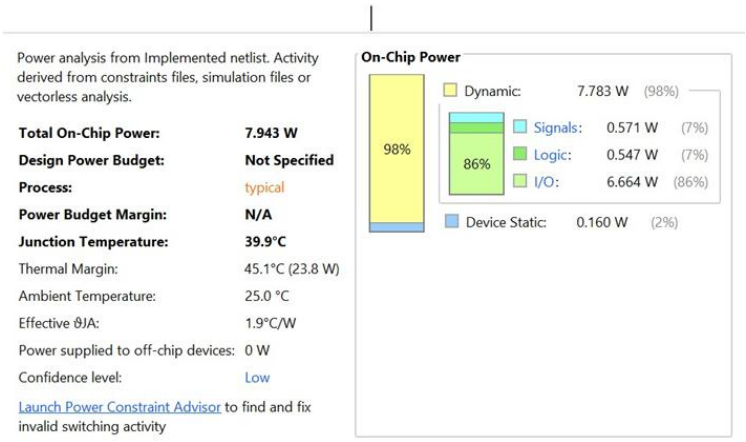
Total On-Chip Power:	**7.998 W**
Design Power Budget:	**Not Specified**
Process:	typical
Power Budget Margin:	**N/A**
Junction Temperature:	**40.0°C**
Thermal Margin:	45.0°C (23.8 W)
Ambient Temperature:	25.0 °C
Effective ϑJA:	1.9°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

Launch Power Constraint Advisor to find and fix invalid switching activity

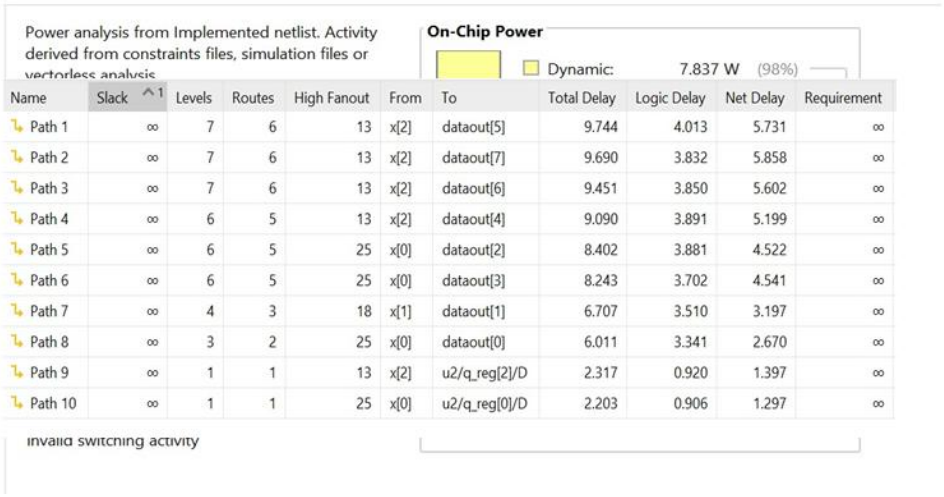


Power Report

APPROXIMATION POWER REPORT:



Time Report



Name	Slack ^1	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement
Path 1	∞	7	6	13	x[2]	dataout[5]	9.744	4.013	5.731	∞
Path 2	∞	7	6	13	x[2]	dataout[7]	9.690	3.832	5.858	∞
Path 3	∞	7	6	13	x[2]	dataout[6]	9.451	3.850	5.602	∞
Path 4	∞	6	5	13	x[2]	dataout[4]	9.090	3.891	5.199	∞
Path 5	∞	6	5	25	x[0]	dataout[2]	8.402	3.881	4.522	∞
Path 6	∞	6	5	25	x[0]	dataout[3]	8.243	3.702	4.541	∞
Path 7	∞	4	3	18	x[1]	dataout[1]	6.707	3.510	3.197	∞
Path 8	∞	3	2	25	x[0]	dataout[0]	6.011	3.341	2.670	∞
Path 9	∞	1	1	13	x[2]	u2/q_reg[2]/D	2.317	0.920	1.397	∞
Path 10	∞	1	1	25	x[0]	u2/q_reg[0]/D	2.203	0.906	1.297	∞

Setup Timing Report

Name	Slack ^1	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay
Path 1	∞	6	5	22	x[0]	dataout[7]	8.870	3.878	4.993
Path 2	∞	6	5	16	x[1]	dataout[6]	8.861	4.110	4.750
Path 3	∞	6	5	16	x[1]	dataout[5]	8.834	4.098	4.736
Path 4	∞	6	5	22	x[0]	dataout[4]	8.493	3.873	4.620
Path 5	∞	6	5	11	x[2]	dataout[2]	8.343	3.898	4.445
Path 6	∞	5	4	11	x[2]	dataout[3]	7.699	3.787	3.913
Path 7	∞	3	2	22	x[0]	dataout[0]	6.383	3.517	2.866
Path 8	∞	3	2	22	x[0]	dataout[1]	6.089	3.385	2.704
Path 9	∞	1	1	16	rst	u4/q_reg[0]/R	2.372	0.913	1.460
Path 10	∞	1	1	16	rst	u4/q_reg[1]/R	2.369	0.913	1.456

APPROXIMATION HOLD TIME:

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay
Path 11	∞	1	1	3	u2/q_reg[1]/C	u3/q_reg[1]/D	0.270	0.141	0.129
Path 12	∞	1	1	6	u2/q_reg[2]/C	u3/q_reg[2]/D	0.307	0.164	0.143
Path 13	∞	1	1	5	u4/q_reg[0]/C	u5/q_reg[0]/D	0.338	0.141	0.197
Path 14	∞	1	1	7	u4/q_reg[2]/C	u5/q_reg[2]/D	0.338	0.141	0.197
Path 15	∞	1	1	5	u2/q_reg[0]/C	u3/q_reg[0]/D	0.347	0.141	0.206
Path 16	∞	1	1	3	u2/q_reg[3]/C	u3/q_reg[3]/D	0.349	0.164	0.185
Path 17	∞	1	1	6	u4/q_reg[3]/C	u5/q_reg[3]/D	0.356	0.141	0.215
Path 18	∞	1	1	7	u3/q_reg[3]/C	u4/q_reg[3]/D	0.421	0.164	0.257
Path 19	∞	1	1	10	u3/q_reg[2]/C	u4/q_reg[2]/D	0.428	0.164	0.264
Path 20	∞	1	1	4	u4/q_reg[1]/C	u5/q_reg[1]/D	0.519	0.141	0.378

Hold Time Report

V. CONCLUSION

The proposed work presents the design and implementation of a low-power approximate FIR filter intended for energy-efficient VLSI systems. FIR filters are widely used in digital signal-processing applications, but their repeated multiplication and accumulation operations can result in significant hardware complexity, power consumption, and processing delay. The project addresses these challenges by incorporating approximation techniques into arithmetic operations and by investigating efficient multiplier and adder architectures. The proposed design employs an array-based multiplier structure together with approximate arithmetic units to reduce unnecessary computational complexity. High-speed addition is achieved using parallel-prefix concepts, with the Ladner–Fischer adder selected as the principal high-performance addition architecture. These architectural choices provide a suitable basis for improving the speed and implementation efficiency of FIR filtering.

The complete design is described using Verilog HDL and evaluated through simulation and synthesis in the Xilinx Vivado environment. The project includes functional simulation, RTL schematic analysis, power estimation, and setup- and hold-time evaluation. The results demonstrate that the approximate FIR filter operates correctly while providing a useful trade-off between computational accuracy and hardware performance. The reported implementation power is 7.998 W for the stated configuration, although a complete numerical comparison with the conventional architecture is not provided in the report. Approximation introduces controlled computational error; nevertheless, this limitation can be acceptable in error-resilient applications such as DSP, image processing, audio processing, IoT devices, and wearable electronics. Overall, the work demonstrates that combining approximate arithmetic with efficient multiplier and high-speed adder structures can provide a practical approach to low-power FIR-filter implementation. Future improvements can focus on further power reduction, area optimization, improved timing, higher filter orders, FPGA/ASIC implementation, and improved accuracy.

REFERENCES

1. Wanhammar, L. (1999). *DSP integrated circuits*. Academic Press.
2. Wallace, C. S. (1964). A suggestion for a fast multiplier. *IEEE Transactions on Electronic Computers, EC-13*(1), 14–17.
3. Gallagher, W. E., & Swartzlander, E. E., Jr. (1994). High radix Booth multipliers using reduced area adder trees. In *Proceedings of the Asilomar Conference on Signals, Systems & Computers* (Vol. 1, pp. 545–549).
4. McClellan, J. H., Parks, T. W., & Rabiner, L. R. (1973). A computer program for designing optimum FIR linear phase digital filters. *IEEE Transactions on Audio and Electroacoustics, 21*(6), 506–526.
5. Nguyen, H., & Chatterjee, A. (2000). Number-splitting with shift-and-add decomposition for power and hardware optimization in linear DSP synthesis. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 8*(4), 419–424.
6. Ercegovac, M. D., & Lang, T. (2003). *Digital arithmetic*. Morgan Kaufmann.
7. Hartley, R. I. (1996). Subexpression sharing in filters using canonic signed digit multipliers. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, 43*(10), 677–688.
8. Park, I.-C., & Kang, H.-J. (2001). Digital filter synthesis based on minimal signed digit representation. In *Proceedings of the 38th Design Automation Conference* (pp. 468–473).
9. Aksoy, L., Costa, E., Flores, P., & Monteiro, J. (2008). Exact and approximate algorithms for the optimization of area and delay in multiple constant multiplications. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 27*(6), 1013–1026.
10. Dempster, A. G., & Macleod, M. D. (1995). Use of minimum-adder multiplier blocks in FIR digital filters. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, 42*(9), 569–577.
11. Voronenko, Y., & Püschel, M. (2007). Multiplierless multiple constant multiplication. *ACM Transactions on Algorithms, 3*(2), 1–39.
12. Aksoy, L., Güneş, E. O., & Flores, P. (2010). Search algorithms for the multiple constant multiplications problem: Exact and approximate. *Microprocessors and Microsystems, 34*(5), 151–162.
13. Hartley, R., & Parhi, K. K. (1995). *Digit-serial computation*. Kluwer Academic Publishers.
14. Ladner, R. E., & Fischer, M. J. (1980). Parallel prefix computation. *Journal of the ACM, 27*(4), 831–838.
15. Brent, R. P., & Kung, H. T. (1982). A regular layout for parallel adders. *IEEE Transactions on Computers, C-31*(3), 260–264.
16. Erdogan, A. T., & Arslan, T. (2002). On the low-power implementation of FIR filtering structures on single multiplier DSPs. *Microelectronics Journal, 33*(3), 223–229.

17. Venkata Sivaiah, U., Prasanna Murali Krishnna, P., & Devaraju, Y. (2013). Implementation of low power digital FIR filter design based on low power multipliers and adders. *International Journal of Computer Sciences and Engineering*.

18. Deepika, A., & Bhuvaneswari, A. (2014). Low power FIR filter design using truncated multiplier. *International Journal of Engineering Trends and Technology, 10*(1), 34–39.

19. Ye, W. B., Lou, X., & Yu, Y. J. (2017). Design of low power multiplierless linear-phase FIR filters. *IEEE Access*.

20. Kumar, A., Sharma, I., & Balyan, L. K. (2021). Design of low power multiplierless FIR filter with enhanced adder efficiency using flower pollination optimization. *Applied Acoustics, 174*, 107792.

21. Farshchi, F., Abrishami, M. S., & Fakhraie, S. M. (2020). New approximate multiplier for low power digital signal processing. *arXiv*.

22. Jiang, H., Liu, C., Liu, L., Lombardi, F., & Han, J. (2017). A review, classification, and comparative evaluation of approximate arithmetic circuits. *ACM Journal on Emerging Technologies in Computing Systems, 13*(4).

23. Jiang, H., Santiago, F. J. H., Mo, H., Liu, L., & Han, J. (2020). Approximate arithmetic circuits: A survey, characterization, and recent applications. *Proceedings of the IEEE*.

24. Kumm, M., Volkova, A., & Filip, S.-I. (2019). Design of optimal multiplierless FIR filters. *arXiv*.

25. Nair, A. R., Nath, P. K., Chakrabartty, S., & Thakur, C. S. (2023). Multiplierless in-filter computing for tinyML platforms. *arXiv*.

26. Ranjbar, A., Esmaeili, E., Rafieisangari, R., & Shiri, N. (2025). FPGA-based multiplier with a new approximate full adder for error-resilient applications. *arXiv*.

27. Han, J., & Orshansky, M. (2013). Approximate computing: An emerging paradigm for energy-efficient design. In *2013 18th IEEE European Test Symposium*.

28. Venkataramani, S., Choi, J., Choi, Y., & others. (2015). Quality programmable vector processors for approximate computing. *Proceedings of the International Symposium on Computer Architecture*.

29. Zhu, N., Goh, W. L., & Yeo, K. S. (2010). An enhanced low-power high-speed adder for error-tolerant application. In *Proceedings of the 12th International Symposium on Integrated Circuits*.

30. Mrazek, V., Sarwar, S. S., Sekanina, L., Vasicek, Z., & Roy, K. (2016). Design of power-efficient approximate multipliers for approximate artificial neural networks. *Proceedings of the International Conference on Computer-Aided Design*.