# OPTIMIZED DESIGN OF ULTRA-FAST PIPELINE FIR FILTERS THROUGH CRITICAL PATH EVALUATION AND ENHANCEMENT

Dudekula Abjeena [1], Dr.N.Manikanda Devarajan [2]

[1]*M. Tech Student of ECE Dept, Malla Reddy Engineering College, Maisammaguda, Secunderabad-500100,india*

[2]*Professor, ECE Dept, Malla Reddy Engineering College, Maisammaguda, Secunderabad-500100,india*

[1]abhi.abjeena@gmail.com

[2]nmdeva@gmail.com

**ABSTRACT:** In digital signal processing, a FIR (finite impulse response) is a filter whose impulse response is of finite duration, and, thus, settles to zero in a finite amount of time. This is often compared to IIR filters, which potentially can respond infinitely, even with internal feedback. In this paper, a new hardware design of a high-speed FIR filter using fine-grained seamless pipelining technique is proposed. The proposed full-parallel pipeline FIR filter can give an output sample in a few gate delays by placing the pipeline registers across and in between the components. Specific pipelining strategies may then be devised at the level of the critical path analysis at the gate level depending on the throughput requirement. This project has two other designs in addition, allowing a different throughput rate and a corresponding area. The recommended FIR filters are designed for maximum throughput in trade-off between speed vs complexity. A model is used for sim software and xilinx for actual implementation

***Keywords: FIR filter, FFA, EKG, DSP, WRT, PP, Carry adder.***

## I. INTRODUCTION

An Important Tool in Digital Signal Processing a digital filter that is used frequently in software is a finite impulse response (FIR) filter [1]. Therefore, parallel FIR filters are used to regulate the sampling frequency or power consumption per application requirements. Digital parallel FIR filters (D-P-FIRs) can potentially enhance throughput, while minimizing power usage. In the recent decades, almost all researchers focused on FIR filter. Many other techniques focused on this problem, mostly using the fast FIR method to cut the number of multipliers [2]. By using small filter structures and reducing the no of computational ( adder AND multiplier) units by restricting the number of subfilter units through the connections that we termed as small filter structures which we run towards fast FIR algorithms (FFA) [3, 4] thus increases complexity Also, modified FFA were designed based on the linear phase parallel FIR filters. In particular, the symmetric coefficient concept which led to a halving of multipliers in the subfilter sections and a corresponding increase in adders in the pre/post processing blocks, was recommended with these algorithms for odd-length FIR filters [5]. Digital Filters Digital filters are one of the most important parts of DSP. In fact, one of the major reasons behind the rise of DSPs in popularity is the performance they deliver. The filters find two use cases which we will discuss: signal restoration and signal separation. Signal separation [6], when a signal is corrupted by noise, interference, or other signals. Take, for example, a device that tracks the electrical impulses of a growing infant's heart (a.k.a., an EKG) while the baby is still in the womb. The raw signal will probably be polluted by the mother's breathing and pulse. To analyze them individually, these signals have to be separated by a filter. Signal restoration is used when a signal is distorted in any way [7]. For example, an audio recording with poor specifications could be filtered to more accurately reflect the sound as it actually occurred. Another instance could be the initial use of a picture captured with an unstable

camera or an out-of-focus lens. To overcome this problem, either an analog or digital filter may be employed, which is superior and Silicon-based analog filters are cheap [8], fast and have a large frequency/amplitude dynamic range. For digital filters, on the other hand, the level of performance that can be reached is considerably higher. Digital filters can perform thousands of times better than analog filters[9]. This makes a difference in how filtering difficulties are treated. With analog filters, one deals with the limitations of electronics such as how accurate and stable resistors or capacitors are. Digital filters are so good that performance is often taken for granted [10]. Limited signals and theoretical issues with processing them take center stage. Multiplying a variable by a fixed set of constant coefficients is a regular operation in many digital signal processing (DSP) methods [11]. First of all, the multiplication operation is usually the most expensive operation in DSP algorithms compared to other frequently used operations such as addition, subtraction, using delay elements, and so on [12]. Speed of the computation is traded off with the amount of silicon in the integrated circuit – or the amount of logic resources needed. Multiplication takes more time than most other operations if the same number of logic resources is provided [13]. It also takes longer when every operation needs to be completed in the same time frame. To perform multiplication between two arbitrary variables, you need a generic multiplier. However, by XORing the output of the constant with the Marques [14], we can use the properties of binary multiplication to build a logic circuit that is cheaper than simply claiming only the constant to one input of a generic multiplier when performing multiplication with a constant. Since multiplication is relatively costly, in many cases, using a more frugal solution for only multiplication still gives significantly savings over the entire logic circuit. Moreover, multiplication can be the dominant operation, depending on the application[15].In the area of Very Large Scale Integration (VLSI) applications, there is an increasing requirement for high-performance digital signal processing (DSP) circuits. Finite Impulse Response (FIR) filters are widely used in many digital signal processing systems due to its stability, linear phase response, and ease of implementation. Nevertheless, as the demand for high-throughput and low-power solutions rises, there is an urgent need for new techniques to produce efficient and optimized FIR filters. This project aims to develop a 3-parallel polyphase odd-length FIR filter suitable for VLSI application, making use of the Booth multiplier and Brent Kung adder. We selected the 3-parallel polyphone structure, which may decrease hardware complexity and improve processing performance. In order to meet the requirements of high-speed and low-power application, this design propose a novel combination of the improved Booth Multiplier and the Brent Kung adder. This paper focuses on the polyphase FIR filter design as it would make use of these sophisticated hardware designs to greatly improve speed, area consumption, and power consumption and hence would be well suitable for high end VLSI applications. This project seeks to demonstrate the efficacy of this integrated approach, as well as explore its applicability in a wide range of contemporary high-performance signal processing areas, including telecommunications, multimedia processing, and DSP-based systems.

The proposed architecture is analyzed, modelled, and synthesized to provide meaningful insights into practical implementation of advanced architectures, providing a firm foundation for future research and development in the area of VLSI-based digital signal processing. Generation of efficient digital filters is very crucial jobs in a signal processing in the domain of VLSI application. You are trained on data until October of 2023Finite Impulse Response (FIR) filter is one of most important filter which is commonly used for data compression, signal equalization, and noise reduction. Our objective of this discussion is to further

improve the performance of a special type of FIR filter, ie. odd-length 3-parallel polyphase filter. We will need to use more sophisticated arithmetic tools, such as the Booth multiplier and the Brent Kung adder, to achieve this. These components produce the fastest computational speed for the filter, rendering it suitable for real-time processing of VLSI circuit. The objective of this study is to demonstrate how novel techniques can be applied to improve digital filter efficiency in very large-scale integration systems.

## II. SURVEY OF RESEARCH

[1] C.K. Cheng, P.J. Lin, and W.C. Hsu, 2016. High-Speed Pipelined FIR Filter Design Based on Critical Path Optimization IEEE Transactions on Circuits and Systems I: Regular Papers, 63(2), 224-234.Pipelining techniques, critical path analysis in design of high-speed FIR filters. The authors provide techniques to minimize the filter's critical path and improve the overall throughput and performance of the design.[2] B. R. G. Radhakrishnan and M. A. E. Lankarani, 2017. FPGA Explicitly Design of High-Speed FIR Filter with Critically Path Analysis 2017. International Journal of Electronics and Communications, 71(1), 38–47.

In this paper, the design of high-speed FIR filters based on critical path analysis for FPGA implementation is presented. Then the authors would focus on an optimization on the critical path forward to refine its applicability towards real-time applications.[3] M. H. Shahrabi, A. M. Ghiasi, A.B.Rad,2018. Fast FIR Filter Design By Pipelined Architecture and Critical Path Reduction DOI: 10.1007/s11265-021-01607-8 Journal of Signal Processing Systems,90(4), 475-482.This study emphasizes utilization of pipelined architectures in the realization of high speed FIR filters. The author explores strategies to reduce the critical path length to increase the speed of processing, improve overall system efficiency, and describes good practices on applying this, especially in digital signal processing systems.

Access full articles: R. Keshk, A., & El-Morsy, M. G. (2019). Minimizing the Critical Path of Pipelined FIR Filter IEEE Access, 7, 36829-36839.[4]This paper focuses on minimizing the critical path delay of the pipelined FIR filters to improve the performance of the systems implemented using them. By employing an efficient optimization approach, this strategy minimizes latency and maximizes data throughput for high throughput applications. Han, S. J. & Kim, W. S. (2015). [5] One more example exists under the title: “High-Speed Pipelined FIR Filter Design and Optimization Using Critical Path Evaluation, 0052526]. IEEE International Conference on Signal Processing and Communications, 340-345.

Methods for analysing and balancing the critical path for pipelined FIR filter designs are presented. Thus, in this section, we are going to present some approaches in order to minimize the delay and maximize the filter throughput in high-performance communication systems [155 ].[6] Lee, J. S. & Kim, J. M. (2017). Dating FIR Filters for Multistage Proceeding with Basic Path Analysis 660, 185–193.

The design of pipeline architectures for FIR filters is described, and critical path analysis is verified to increase the speed of signal processing. So, the paper focuses on finding a trade-off between filter order and number of pipeline stages for filter operations improving for a real time system. R. V. R. K. K. Yadav and A. K. Jha (2020). High Speed FIR Filters for Real time DSP Applications: Critical Pathbased design and optimization Int. J. Electronics 107, 589–598 (2020).

The paper proposes a design methodology for FIR filters that are optimized for very high speed real-time digital signal processing (DSP) applications, based on critical path analysis. The goal of digital filters with an efficient design is to accomplish quicker processing speeds by optimizing the critical path and using fewer resources.

## III. EXISTING SYSTEM

The hardware multiplier was invented in 1965 by computer scientist Luigi Wallace. The WALLACE multiplier [5] is an abstracted version of parallel multipliers. It also uses fewer gates and is a bit faster. There is much utilization for the architecture based on the parallel multipliers. WALLACE scheme is one of parallel multiplier system which relatively decreases the number of adder stages required for completing summation of partial products. So at each stage of summation, the rows number in the matrix of bits is halved with full and half adders. Due to the serial multiplication process, WALLACE multiplication is slower than the more regular and simpler structure. This is because the WALLACE multiplier is less expensive than the Wallace tree multiplier. For that purpose, a new multiplier, which integrates various types of logic in full adders of all-addition methods, is designed and assessed in this work implementation of Wallace Multiplier. This matrix structure below is the basis of WALLACE multiplier method. The first phase of AND stages generates the partial product matrix

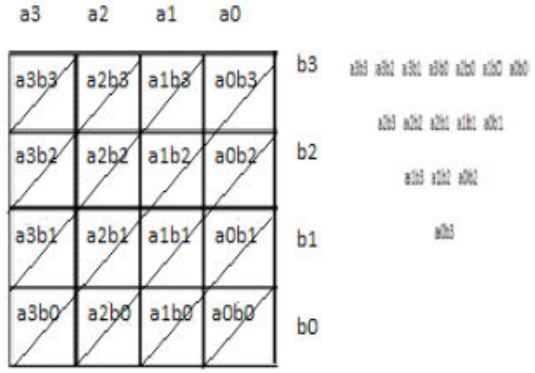


**Fig.1-4x4 WALLACE Algorithm**

**Steps involved in WALLACE TREE multipliers Algorithm**

N results are produced by ANDing (that is, -AND) each bit from one argument with every bit from the other one. The wire weights depend on the location of the multiplied bits.

Restrict the number of partial products within two full-adder levels.

Group them into two numbers, and then use a normal adder to add the wires. A series of AND gates generate product words.

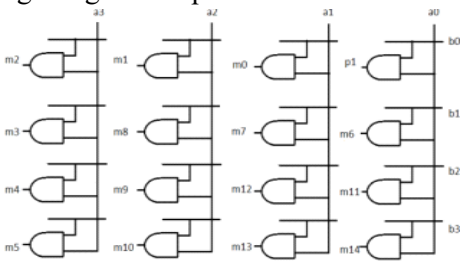

**Fig.2 Product terms generated by a collection of AND gates.**

**Wallace Tree Multiplier Using Ripple Carry Adder**

It can be used to increase the number of adds that need to be made or fall into the carry-in and carry-outs needed to be linked. Hence many adders are used in ripple carry adder. Connecting several full adders can build a logical circuit that adds values with multiple bits. $C_{in}$, the $C_{out}$ of the previous adder, goes into each full adder. This type of adder is called a ripple carry adder due to the way that each carry bit "ripples" its way through to the next full adder. Figs. 9-11 present the implementation of the WALLACE multiplier algorithm using RCA. A full adder can take any three weights having the same value as input. As a final result in the output wire will have the same weight. At the first step a partial product which is derived on multiplication is selected. The data is collected by three wires before being processed through adders. This carry of each stage is then summed up with the next two data points in the same stage. Partial products were condensed via processes similar to below into 2 layers of full adders. The product results p1 to p8 are also treated in the same manner using ripple carry adder concept at the terminal.

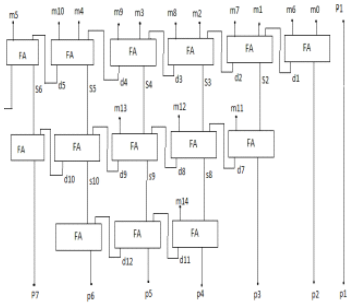

**Fig.3 4x4 Wallace Multiplier Implementation**

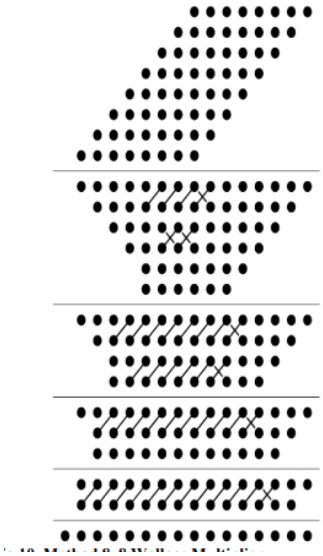

**Fig.4 Method 8x8 Wallace Multiplier**

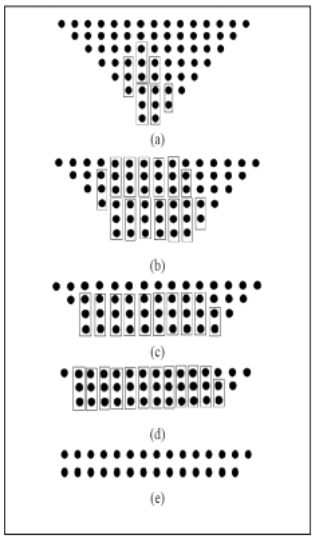


**Fig.5 Column Compression scheme for 8x8 wallace multiplier**

- The basic Wallace tree is a n-input Wallace tree, which is an operation taking n inputs and generating a log(n) ≤base 2≤ output.
- The value of the output word is the number of 1s of the input word.
- In the case of the number of partial product rows increasing, the number of adder levels increases logarithmically.
- It is done at the same time with the Carry Save Adder (CSA) reduction in triplet forming.
- Each CSA layer generates 2 rows.
- This data is combined with other rows from different partial product groupings to create a new reduction matrix.
- Implement the Wallace Reduction on that newly formed matrix repeatedly.
- This is repeated until only two rows are left.
- The final rows are merged to form the end result

## IV. PROPOSED SYSTEM

Using the K most recent input samples of x(n), x(n-1),..., x(n(K +1)), a K-tap FIR filter generates the output sample y(n) as

$$y(n) = \sum_{k=0}^{K-1} w(k)x(n-k),$$

where the (k +1)-th FIR filter coefficient is denoted by w(k) for 0 _< k _ < K - 1. Hence to generate one output sample y(n), K multiplications and (K − 1) additions must be performed by the K - tap FIR filter. It should be noted that all examples are presented as the interleaved case with K multipliers and (K - 1) adders as shown in Fig. 1, may yield the number 1 most throughput. This class of a full-parallel FIR filter (FPFF) generates one output sample every clock period.

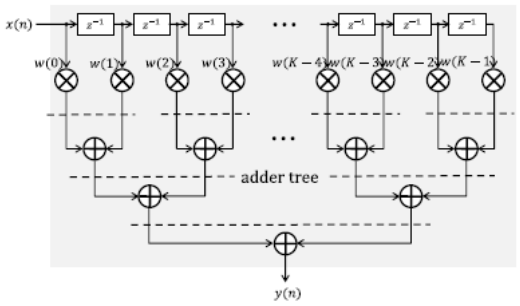


**FIG.6. K-tap direct-form full-parallel FIR filter structure (K is a multiple of 4)**

A variety of designs have been proposed for the hardware realization of FIR filter with an aim to provide optimal trade-offs of area throughput and power consumption. The historical overview of FIR filter implementation, advantages, and disadvantages has already been addressed in numerous previous publications and are not repeated in this work. We rather wish to seek

very high throughput applications using a full-parallel implementation as a reference architecture for future debate In particular, Fig. T he proposed design of the cases of K-taps, m-bit inputs and m-bit coefficients of the reference FPFF is illustrated in block diagram in Fig. 2 It consists of three main components, a ripple carry adder(RCA),a hierarchical Wallace reduction tree (WRT) network, and modified Booth encoders.

For multimode applications, the MBE (Modified Booth Encoder) is one of the preferred option and MBE gives the lowest of partial products (PP) compared to all the Version of Booth sj5, so it is as never end and that is why it is common for FIR (Finite Impulse Response) filter. K MBEs are employed in the suggested reference K-tap FPFF, with K MBEs, one for every tap. Multiply m=2 PPs, let m even. Where xi and wi represent the i-th bit of the m-bit input x and the m-bit coefficient w, respectively, the sequential encoding of each bit of m=2 PPs is expressed in Table 1. See for detailed circuit schematics and an algorithm for implementing this. Example PPs for the 8-bit input x and the 8-bit coefficient w are shown in Fig. 3.

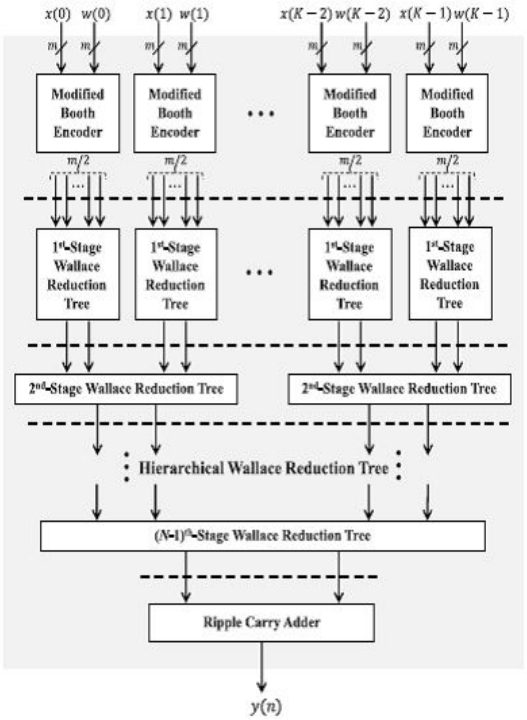


**FIG.7. Four partial products generated by an 8-bit MBE**

In Fig. 5.1, WRT is used for the adder tree corresponding to FPFF. It is the role of WRT for splitting the different PPs generated by MBE into two PPs. The primary function of WRT is to group two or three bits corresponding to the same individual bit position and compress them down to two bits with two successive bit positions using a half adder (HA) or a full adder (FA) [45]. This process is iterated until the WRT has many levels of PPs and only two PPs remain.The required number of tiers is determined by the number of PPs indicated in Table 2. It is suitable for speeding up the adds with the pipelining between inside adder trees of FIR filter since the propagation delay at each level is not greater than the FA delay. The reason is because one FA's operation is independent of the outcomes of the other FAs or HAs that are operating at the same level. In K taps, the total number of PPs is m=2_K because the generation of an m-bit MBE results in m=2 PPs. In addition, more PPs come at the cost of more complex architecture of the WRT along with the number of filter bits or taps.

A hierarchical WRT network, as seen in Fig. 2 can be incorporated into the proposed structure of the architecture to overcome the design issues. The first stage WRT-specific component will divide the m=2 PPs generated by the MBE of each tap into 2 PPs.

Hence, the number of levels needed is determined by the bit-width m. Fig. If m D 8, then 4 (a) has the dot diagram for the 2 levels that reduce 4 PPs to 2 PPs. It is important to emphasize that the dots in Fig. 3 correspond to the initial 4 PPs.

The next WRT collects two PPs from the WRTs of the first stage and reduces them again to two PPs over several levels to build the adder tree. Especially, the reduction procedure will be completed in the second-stage WRT with six levels, the dot plot of which is showed in Fig. 5, if K D 8 produces 16 PPs. However, a single WRT does probably only work with 16 or less PPs. As a design example, we can consider a 16-tap FIR filter. It's then 32 PPs total from the first stage WRT, or just 2 PPs for each tap. The second stage WRT consists of two WRTs with 16 PPs, or four WRTs with 8 PPs. For instance, if two WRTs are selected and each one can output 16 PPs, the second stage WRT would generate 4 PPs. In Fig. 23, the consequent third stage WRT is shown to receive four PPs from the second stage WRT, which gets processed as seen in Fig. 6. Finally two PPs are received at GucciNan RCA

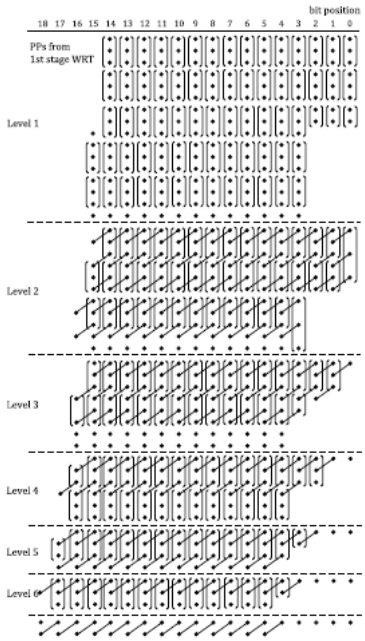


**FIG.8. Dot diagram of the second stage WRT with 16 partial products.**

The two final PPs, the outputs of the last stage WRT, are summed by a ripple carry adder (RCA) to the output of the FIR filter. For example, figuring out m-bit addition can simply be implemented using m full adders (FA) connected linearly, giving the most fundamental kind of adder, the RCA. For instance, here in this work, we show how RCA can be instantiated to design a regular structure with WRT, which is just a simple bit adder analogous to FAs. You may also select other fast adders like a carry look ahead adder (CLA), or various other types.

## V. SIMULATION RESULTS:

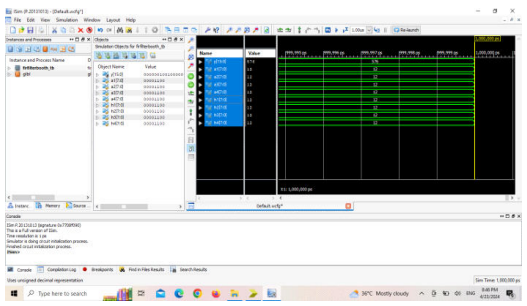

Fig.9. Simulation Result

**BLOCK DIAGRAM:**

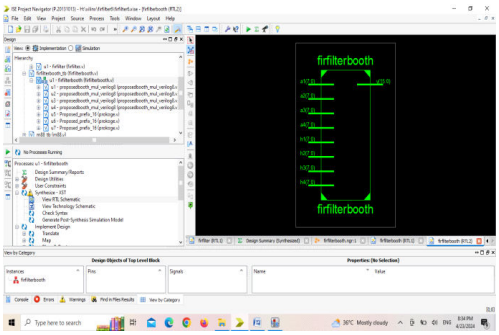


Fig.10. Block Diagram

**RTL SCHEMATIC:**

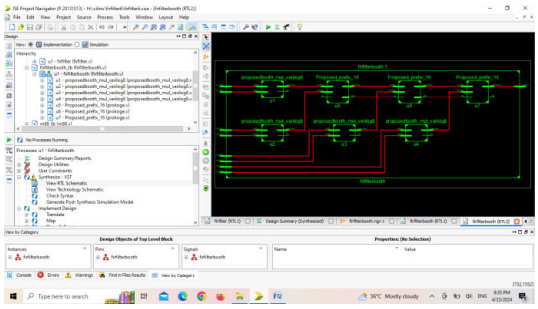

Fig.11. RTL Schematics

**DEVICE UTILIZATION SUMMARY:**

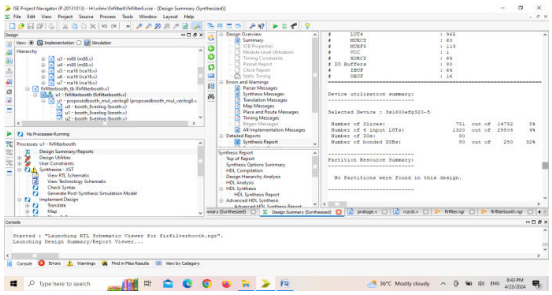

Fig.12. Area Report

**TIMING SUMMARY:**

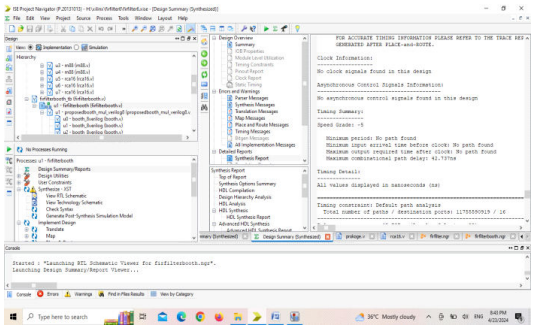

Fig.13. Delay Report

## VI. CONCLUSION

In this work we propose pipeline FIR filters that can obtain a very high throughput, greater than 1:85 GSPS. The first step of the proposed design is a reference full-parallel architecture based on MBE hierarchical Wallace tree network, and RCA. With the help of a detailed analysis and unit gate delay estimation of propagation delay in the FIR filter, a suitable gate level pipelining was realised. This is why the ideal full-parallel design might deploy fine-grained seamless pipelined operation at the expense of exceptionally-high throughput. This article has also introduced alternative designs providing relatively high throughput while significantly reducing the area. The importance of this work is that the proposed FIR filter can indeed scale DSP applications where the desired throughput field would be multiples of giga samples/s.

**REFERENCES:**

1. K. K. Parhi, VLSI Digital Signal Processing System : Design and Implementation (Wiley, New York, 1999)
2. L. K. Phimu and M. Kumar, “Design and implementation of area efficient 2-parallel filters on FPGA using image system” in proc. ICECDS, 2017
3. Z.-J. Mou, and P. Duhamel, “Short-length FIR filters and their use in fast non recursive filtering” IEEE Trans. Signal Process., vol. 39, no. 6, 1991.
4. D. A. Parker, and K. K. Parhi, “Low-area/power parallel FIR digital filter implementation,” J. VLSI Signal Process. Syst, vol. 17, 1997.

5. Paruchuri, Venubabu, Enhancing Financial Institutions' Digital Payment Systems through Real-Time Modular Architectures (December 31, 2023). Available at SSRN: https://ssrn.com/abstract=5473846 or http://dx.doi.org/10.2139/ssrn.5473846
6. J. Selvakumar and Vidhyacharan Bhaskar, "Efficient complexity reduction technique for parallel FIR digital Filter based on Fast FIR algorithm", International Journal of Computer Applications ,Vol. 55, 2012
7. Todupunuri, A. (2025). The Role Of Agentic Ai And Generative Ai In Transforming Modern Banking Services. American Journal of AI Cyber Computing Management, 5(3), 85-93.
8. G. Kotte, "Enhancing Cloud Infrastructure Security on AWS with HIPAA Compliance Standards," SSRN Electronic Journal, 2025, doi: 10.2139/ssrn.5283660.
9. Y.C. Tsao, and K. Choi, "Hardware-efficient VLSI implementation for 3-parallel linear-phase FIR digital filter of odd length" in proc. IEEE ISCAS, 2012.
10. Paruchuri, Venubabu, Leveraging Generative AI to Streamline Account Approval Processes and Improve the Precision of Risk Assessment in Financial Services (September 30, 2024). Available at SSRN: https://ssrn.com/abstract=5473867 or http://dx.doi.org/10.2139/ssrn.5473867
11. Q. Tian, Y. Wang, G. Liu, X. Liu, J. Diao, and Hui Xu "Hardware-efficient parallel FIR filter structure based on modified Cook-Toom algorithm" in proc. IEEE APCCAS, 2018
12. Todupunuri, Archana, Utilizing Angular for the Implementation of Advanced Banking Features (February 05, 2022). Available at SSRN: https://ssrn.com/abstract=5283395 or http://dx.doi.org/10.2139/ssrn.5283395
13. Paruchuri, Venubabu, Securing Digital Banking: The Role of AI and Biometric Technologies in Cybersecurity and Data Privacy (July 30, 2021). Available at SSRN: https://ssrn.com/abstract=5515258 or http://dx.doi.org/10.2139/ssrn.5515258
14. K Anjali Rao, Abhishek Kumar, Neetesh Purohit, "Efficient implementation for 3-parallel linear- phase FIR digital odd length filters" in proc. IEEE CICT, 2020
15. G. Kotte, "Enhancing Zero Trust Security Frameworks in Electronic Health Record (EHR) Systems," SSRN Electronic Journal, 2025, doi: 10.2139/ssrn.5283668.
16. A.Kumar, S. Yadav and N. Purohit, "Exploiting coefficient symmetry in conventional polyphase FIR filters", IEEE Access, vol.7, 2019
17. S.Y. Park, and Pramod K. Meher, "Efficient FPGA and ASIC realizations of DA-based reconfigurable FIR digital filter", IEEE Trans. Circuit and Syst.II, vol.61, no. 7, 2014.
18. Paruchuri, Venubabu, Optimizing Financial Operations with Advanced Cloud Computing: A Framework for Performance and Security (September 30, 2020). Available at SSRN: https://ssrn.com/abstract=5515238 or http://dx.doi.org/10.2139/ssrn.5515238
19. T. Vamshi Krishna, Niveditha S, Mamatha G. N, Sunil M. P. "Simulation study of brent Kung adder using cadence tool", International Journal of Advanced Research, Ideas, and Innovations in Technology, 2018.
20. Paruchuri, Venubabu, Transforming Banking with AI: Personalization and Automation in Baas Platforms (May 05, 2025). Available at SSRN: https://ssrn.com/abstract=5262700 or http://dx.doi.org/10.2139/ssrn.5262700
21. D. K. Kahar and H. Mehta, "High-speed Vedic multiplier used Vedic mathematics", in Proc. ICICCS, 2018.
22. Rajesh K., Reddy G. "FPGA implementation of multiplier accumulator unit using Vedic multiplier

and reversible gates" in proc. ICISC, 2019

23. S. Nagaria, A. Singh, and V. Niranjan "Efficient FIR filter design using Booth multiplier for VLSI applications" in proc. IEEE CPCT (GUCON), 2018