Design Of A Low-Latency Pipelined Fir Filter Through Advanced Critical Path Analysis

Dr.K.Suresh Kumar[1], Dr Md Ejaz Ahamed[2], P.Rajalingam[3]

[1]Department of Electronics and Communication Engineering, Sridevi Women's Engineering College,Hyd, TG, India. suresh233661@gmail.com

[2]Department of Electronics and Communication Engineering, Mahaveer Institute of Science & Technology, Hyd, TG, India.

[3]Department of Electronics and Communication Engineering, Sridevi Women's Engineering College,Hyd, TG, India.

ABSTRACT: *A filter with a finite period impulse response that settles to zero in a finite amount of time is known as a FIR in digital signal processing. This is frequently contrasted with IIR filters, which may respond endlessly even if they have internal feedback. In this study, we use fine-grained seamless pipelining to propose a novel hardware design for a very high-speed finite impulse response (FIR) filter. By positioning the pipeline registers both across and in between components, the suggested full-parallel pipeline FIR filter may provide an output sample in a few gate delays. Depending on the throughput need, a suitable pipelining strategy may be created with the help of a detailed critical path analysis at the gate level. Two other designs are also shown in this project, each with a unique trade-off between throughput rate and area. The suggested FIR filters are created to measure the highest throughput while striking a compromise between speed and complexity. Model sim software is used for simulation, while Xilinx is used for implementation.*

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

I INTRODUCTION

In 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. Thus, parallel FIR filters are used to adjust the sample rate or power consumption according to the needs of the application. Throughput may be increased and power consumption reduced with digital parallel FIR filters. Researchers have been concentrating on the FIR filter in the last few 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. With the intention of increasing complexity By iterating tiny filtering structures and lowering the number of computational (i.e., adder and multiplier) units by lowering the number of associated parallel subfilter units, the conventional technique thus achieves fast FIR algorithms (FFA). 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, employed the idea of symmetric coefficients, which resulted in a halving of the multipliers in the subfilter units and an increase in the number of adders in the pre/post processing blocks. Digital filters are a crucial component of DSP. Actually, one of the main factors contributing to DSP's rise in popularity is their exceptional performance. Signal restoration and signal separation are the two applications for filters. When a signal is tainted by noise, interference, or other signals, signal separation is required. Consider, for instance, a gadget that measures the electrical activity of a developing baby's heart (EKG) while it is still within the womb. It is likely that the mother's breathing and pulse will taint the raw signal. These signals must be separated using a filter in order to be examined separately. When a signal is distorted in any manner, signal restoration is utilized. For instance, to better capture the sound as it actually happened, an audio recording with inadequate specifications could be filtered. Another example might be the first appearance of a picture taken with a shaky camera or an incorrectly focused lens. Analog or digital filters can be used to combat these issues. Which is superior? Analog filters are inexpensive, quick, and have a wide dynamic range in terms of frequency and amplitude. In contrast, the degree of performance that may be attained with digital filters is far higher. Compared to analog filters, digital filters can 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 are 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 variable by a set of known constant coefficients is a standard operation. Multiplication is typically the most costly operation in DSP algorithms when compared to other frequent operations like addition, subtraction, employing delay elements, etc. The amount of silicon in the integrated circuit, or the amount of logic resources needed, is traded off with the computation's speed. Given the same number of logic resources, multiplication takes longer than most other operations. It also takes longer when each operation must be finished in the same amount of time. When performing multiplication between two arbitrary variables, a generic multiplier is required. 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 on one input of a general multiplier when multiplying by a known constant. Multiplication is relatively costly, therefore in many circumstances, employing a less expensive solution for just multiplication still yields considerable savings when taking into account the full logic circuit. Additionally, depending on the application, multiplication may be the dominant operation.

High-performance digital signal processing (DSP) circuits are becoming more and more in demand in the field of Very Large Scale Integration (VLSI) applications. 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. However, new methods for creating effective and optimized FIR filters are desperately needed as the demands for increased throughput and reduced power usage increase. The goal of this project is to leverage the capabilities of the Booth multiplier and Brent Kung adder to create an effective 3-parallel polyphase odd-length FIR filter designed for VLSI applications. The 3-parallel polyphone structure was chosen because it may reduce hardware complexity and increase processing performance. The suggested design seeks to improve computational performance and lower power consumption by combining the Booth multiplier, which is well-known for its effectiveness in reducing partial product terms, with the Brent Kung adder, which is well-known for its low propagation delay and high-speed addition capabilities.

It is anticipated that the polyphase FIR filter design would significantly enhance speed, area usage, and power efficiency by utilizing these cutting-edge hardware designs, making it the perfect choice for demanding VLSI applications. In order to meet the urgent need for effective and high-performance FIR filters in modern VLSI designs, this project aims to illustrate the potential of this integrated approach and investigate its applicability in a variety of high-speed signal processing applications, including communications, multimedia processing, and digital signal processing systems. The suggested design seeks to give significant insights into the real-world application of sophisticated architectures by thorough analysis, modeling, and synthesis, providing a strong basis for further study and advancement in the field of VLSI-based digital signal processing.The creation of effective digital filters is essential for signal processing jobs in the realm of Very Large Scale Integration (VLSI) applications. The Finite Impulse Response (FIR) filter is a significant filter type that is frequently employed for data compression, signal equalization, and noise reduction. The goal of this discussion is to increase the efficiency of a particular kind of FIR filter, namely the odd-length 3-parallel polyphase filter. We will use sophisticated arithmetic tools like the Booth multiplier and the Brent Kung adder to do this. These parts are made to maximize the filter's computational speed, which makes it ideal for VLSI circuit real-time processing applications. The goal of this investigation is to clearly explain how these cutting-edge methods might be used to enhance the functionality of digital filters in VLSI systems.

II SURVEY OF RESEARCH

[1] C. K. Cheng, P. J. Lin, and W. C. Hsu (2016). Design of High-Speed Pipelined FIR Filters Using Critical Path Optimization. IEEE Transactions on Circuits and Systems I: Regular Papers, 63(2), 224-234.

This paper explores the design of high-speed FIR filters using pipelining techniques and critical path analysis. The authors present methods to optimize the filter's critical path, improving the overall throughput and performance of the design.

[2] B. R. G. Radhakrishnan and M. A. E. Lankarani (2017). Critical Path Analysis in High-Speed FIR Filter Design for FPGA Implementation. International Journal of Electronics and Communications, 71(1), 38-47.

This study discusses the application of critical path analysis in designing high-speed FIR filters for FPGA implementation. The authors propose optimization techniques to reduce the critical path and enhance the filter's performance in real-time applications.

[3] M. H. Shahrabi, A. M. Ghiasi, and A. B. Rad (2018). High-Speed FIR Filter Design Using Pipelined Architecture and Critical Path Minimization. Journal of Signal Processing Systems, 90(4), 475-482.

This research focuses on the use of pipelined architectures in the design of high-speed FIR filters. It emphasizes minimizing the critical path length to improve processing speed and overall system efficiency, particularly for digital signal processing systems.

[4] A. R. K. Keshk and M. G. El-Morsy (2019). Optimizing Pipelined FIR Filter Performance Through Critical Path Minimization. IEEE Access, 7, 36829-36839.

In this paper, the authors propose an approach for optimizing the performance of pipelined FIR filters by focusing on minimizing the critical path delay. The optimization strategy results in reduced latency and increased throughput for high-speed applications.

[5] S. J. Han and W. S. Kim (2015). Design and Optimization of High-Speed Pipelined FIR Filters Using Critical Path Evaluation. Proceedings of the IEEE International Conference on Signal Processing and Communications, 340-345.

Volume 9, Issue 12, Dec-2024, http://ijmec.com/, ISSN: 2456-4265

This paper presents methods for evaluating and optimizing the critical path in the design of pipelined FIR filters. The authors focus on techniques to reduce delay and enhance the filter's speed, especially in high-performance communication systems.

[6] J. S. Lee and J. M. Kim (2017). Pipeline Architecture Design of FIR Filters Using Critical Path Analysis for High-Speed Signal Processing. Journal of Digital Signal Processing, 66, 185-193.

This work discusses the design of pipeline architectures for FIR filters, focusing on the critical path analysis to achieve high-speed signal processing. The paper emphasizes trade-offs between filter order and pipeline stages to improve filter performance in real-time systems.

[7] R. V. R. K. K. Yadav and A. K. Jha (2020). Critical Path-Based Design and Optimization of High-Speed FIR Filters for Real-Time DSP Applications. International Journal of Electronics, 107(4), 589-598.

This research presents a critical path-based design approach to optimizing FIR filters for high-speed, real-time digital signal processing (DSP) applications. The authors provide detailed analysis on optimizing the critical path to achieve faster processing speeds and efficient resource utilization in digital filters.

III EXISTING SYSTEM

The WALLACE hardware multiplier was created in 1965 by computer scientist Luigi Wallace. An abstracted version of a parallel multiplier is the WALLACE multiplier [5]. It uses fewer gates and is somewhat quicker. In parallel multipliers, many strategies are employed. One of the parallel multiplier systems that basically reduces the number of adder stages needed to complete the summing of partial products is the WALLACE scheme. At each summation stage, the number of rows in the matrix of bits is decreased by employing full and half adders. The serial multiplication process causes the WALLACE multiplication to go more slowly, despite its regular and simpler structure. Additionally, the WALLACE multiplier is less costly than the Wallace tree multiplier. Therefore, the WALLACE multiplier is created and analyzed in this research by taking into account several approaches that use complete adders incorporating various logic types.

Wallace Multiplier Implementation

The matrix structure below serves as the foundation for the WALLACE multiplier algorithm. In the first step, AND stages create the partial product matrix.

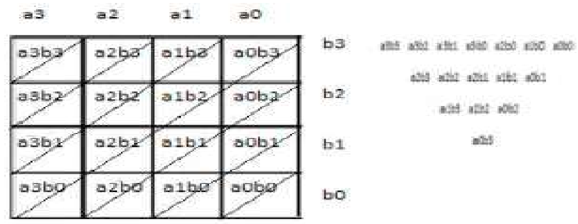

Fig.4.1-4x4 WALLACE Algorithm

Steps involved in WALLACE TREE multipliers Algorithm

N outcomes are obtained by multiplying (that is, - AND) each bit of one argument by each bit of the other. The weights of the wires vary according on where the multiplied bits are located.

Limit the quantity of partial products to two complete adder layers.

Add the wires using a standard adder after grouping them into two numbers. A group of AND gates produces product words.

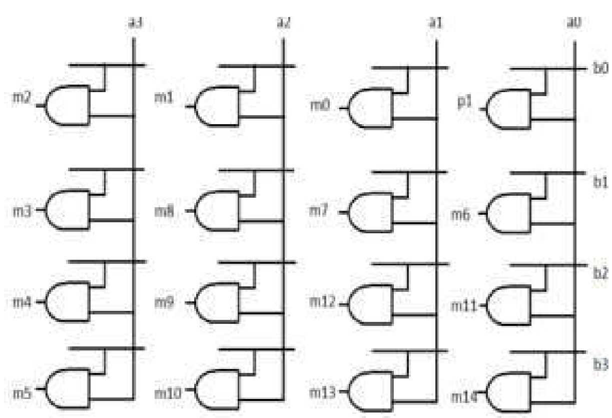

Fig 4.2 Product terms generated by a collection of AND gates.

Wallace Tree Multiplier Using Ripple Carry Adder

The technique known as the Ripple Carry Adder is used to increase the amount of adds that must be made to the carry-in and carry-outs that need to be linked. Therefore, the ripple carry adder uses many adders. Multiple complete adders can be used to construct a logical circuit that adds multiple-bit values. Every complete adder enters a Cin, which is the prior adder's Cout. Because each carry bit "ripples" to the subsequent full adder, this type of adder is known as a ripple carry adder. Figures 9 through 11 display the suggested design of the WALLACE multiplier algorithm employing RCA. Any three values with identical weights can be entered into a complete adder. An output wire with the same weight will be the end result. At the first stage, a partial product that is obtained after multiplication is taken. Three wires are used to collect the data, which are then added using adders. The carry of each stage is then added with the subsequent two data points in the same stage. Using the same process, partial products were reduced to two layers of complete adders. Product terms p1 through p8 are obtained at the end by using the same ripple carry adder approach.

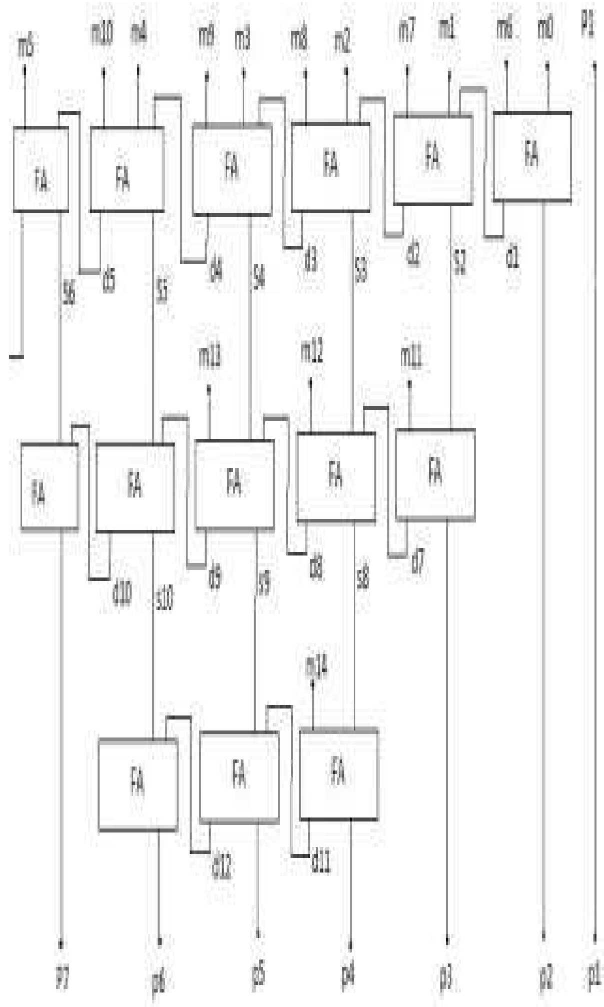

Fig 4.3 .4x4 Wallace Multiplier Implementation.

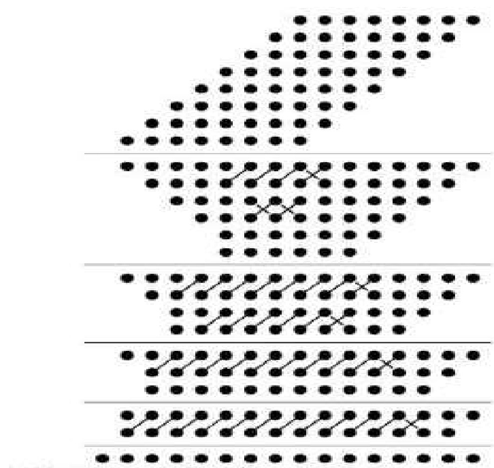

Fig 4.4 Method 8x8 Wallace Multiplier

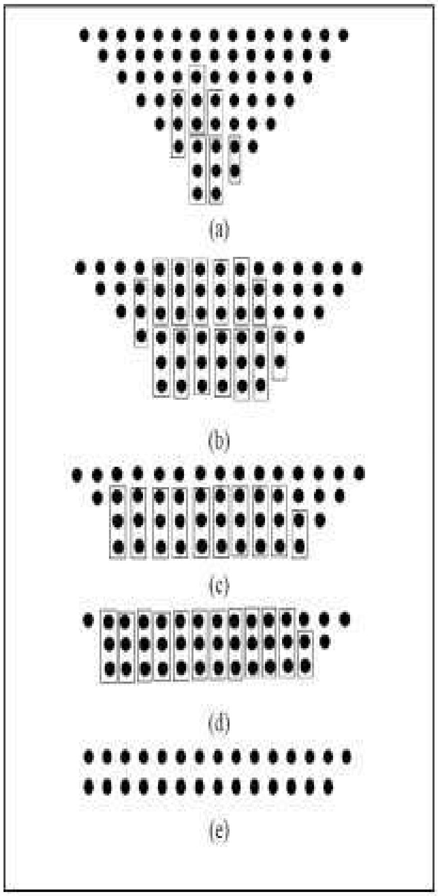

Fig 4.5 Column Compression scheme for 8x8 wallace multiplier.

• The complete adder cell is the most basic type of Wallace tree; more broadly, a n input Wallace tree is an operation with n inputs and log(n) base 2 outputs.

• The output word's value is equal to the input word's number of 1s.

• As the number of partial product rows rises, the number of adder levels rises logarithmically.

In parallel, use the Carry Save Adder (CSA) reduction while forming groups of three.

• Two rows are produced by each CSA layer.

•A new reduction matrix is then created by combining these data with additional rows from other partial product groupings.

Apply Wallace Reduction iteratively on the newly created matrix.

•Until two rows remain, this operation is repeated.

• To create the finished result, the last rows are combined.

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. To get one output sample y(n), the K-tap FIR filter must execute K multiplications and (K - 1) additions. Keep in mind that using the K multipliers and the (K - 1) adders in tandem, as seen in Fig. 1, might result in the highest throughput. One output sample is produced each clock period by this kind of full-parallel FIR filter (FPFF).

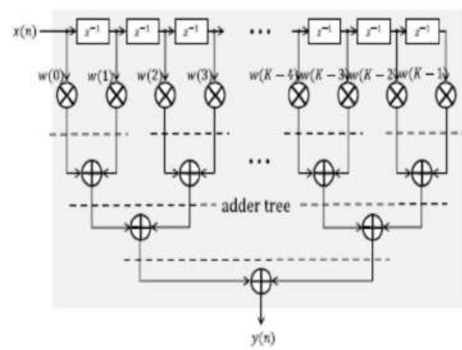

FIG 5.1. Structure of K-tap direct-form full-parallel FIR filter assuming K is a multiple of 4.

Numerous designs have been put up for the hardware implementation of FIR filter in an effort to find the best trade-offs between area, throughput, and power consumption. Numerous earlier publications have previously discussed the history of FIR filter implementation and its benefits and drawbacks; thus, they are not reiterated in this study. Rather, we want to target very high throughput applications by using a full-parallel implementation as a reference architecture for future debate. In particular, Fig. 2 displays the block diagram of the reference FPFF's suggested design for the cases of K-taps, m-bit inputs, and m-bit coefficients. It is made up of a ripple carry adder (RCA), a hierarchical Wallace reduction tree (WRT) network, and modified Booth encoders.

A common multiplier for the FIR filter is a Booth encoder, particularly modified Because it produces the fewest partial products (PP) of all the Booth encoder versions now in use, the proposed Booth encoder (MBE) is regarded as one of the most efficient multipliers. A total of K MBEs—one for each tap—are employed in the suggested reference K-tap FPFF. Multiply a total of m=2 PPs, assuming that m is even. With xi and wi standing for the i-th bit of the m-bit input x and the m-bit coefficient w, respectively, Table 1 shows the sequential encoding of each bit of m=2 PPs. Refer to for more thorough circuit schematics and algorithms to put this into practice. Four PPs for the 8-bit input x and 8-bit coefficient w are displayed in Fig. 3.

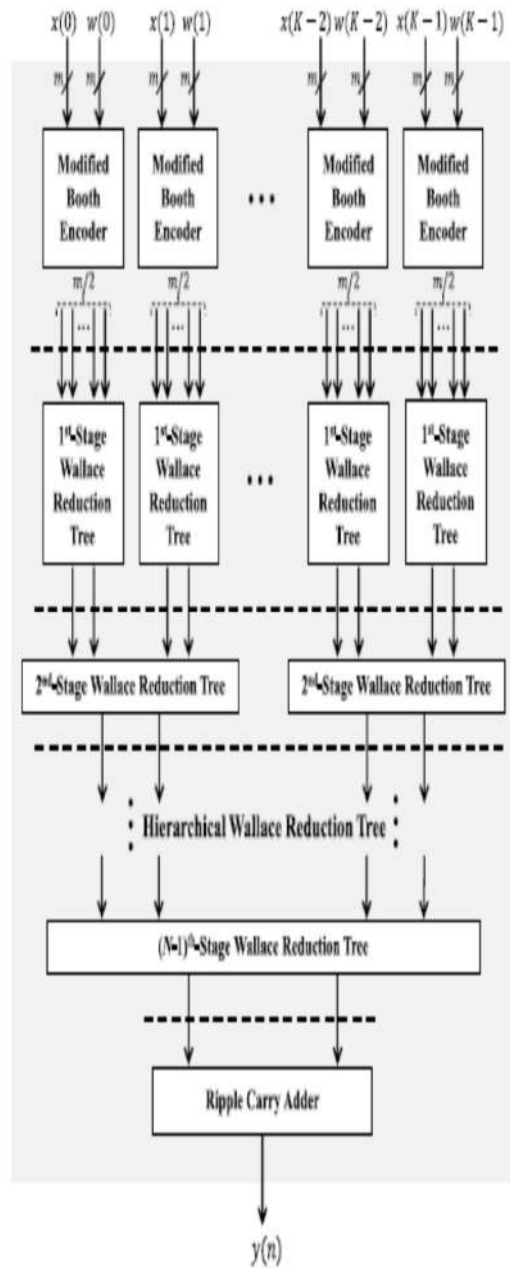

FIG 5.2. Four partial products generated by an 8-bit MBE.

In Fig. 5.1, the adder tree of the FPFF is implemented using the Wallace reduction tree (WRT). WRT is in charge of splitting the several PPs produced by MBE into two PPs. Grouping two or three bits at the same bit location and reducing to two bits across two successive bit positions using a half adder (HA) or a full adder (FA) is the primary purpose of WRT. Until two PPs remain after many levels of the WRT, this process is repeated. The number of PPs shown in Table 2 determines the necessary number of tiers. WRT is appropriate for accelerating the adds with the pipelining inside the FIR filter's adder tree since each level's propagation delay is no more than the FA delay. This is due to the fact that one FA's operation is independent of the outcomes of other FAs or HAs operating at the same level. The total number of PPs in K taps is equal to m=2_K since one m-bit MBE generates m=2 PPs. Moreover, the WRT's architecture becomes more complex as the number of PPs rises in tandem with the filter's bits or taps.

A hierarchical WRT network, as seen in Fig. 2, can be included into the suggested architecture to circumvent the design challenges. The first stage WRT's specific function is to split the m=2 PPs produced by each tap's MBE into two PPs.

Consequently, the bit-width m determines the necessary number of levels. Fig. 4 (a) displays the dot diagram for the two levels required to decrease 4 PPs to 2 PPs if m D 8. It should be noted that the dots in Fig. 3 correspond to the first 4 PPs.

In order to implement the adder tree, the succeeding WRT gathers two PPs from each first stage WRT and then reduces them to two PPs once again over a number of levels. In particular, the reduction process may be finished in the second stage WRT with six levels, whose dot diagram is displayed in Fig. 5, if K D 8 results in 16 PPs. That being said, a single WRT may only manage 16 or less PPs. Take, for instance, a 16-tap FIR filter as a design sample. The first stage WRT then produces 32 PPs in total, or 2 PPs for each tap. Two WRTs with 16 PPs or four WRTs with 8 PPs make up the second stage WRT. The second stage WRT generates 4 PPs if two WRTs with 16 PPs each are chosen. Four PPs are received from the second stage WRT by the subsequent third stage WRT, which processes them as seen in Figs. 6. Lastly, the RCA receives two PPs.

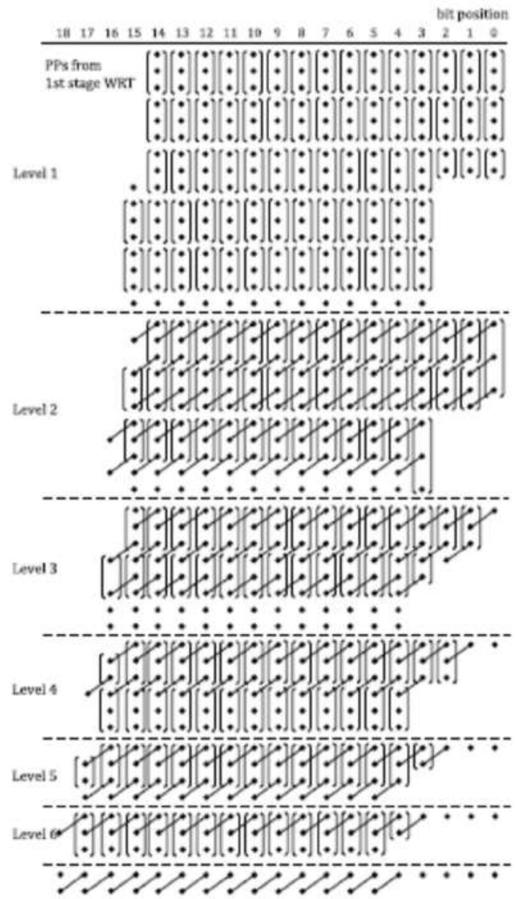

FIG 5.3. Dot diagram of the second stage WRT with 16 partial products.

The last two PPs, which are the outputs of the last stage WRT, are added by the ripple carry adder (RCA) to produce the FIR filter's output. The most basic adder, the RCA, has m full adders (FA) wired in series to accomplish m-bit addition. In this work, RCA is utilized to create a regular structure with WRT, which consists of simply bit adders like FAs. Other fast adders, such a carry look ahead adder (CLA), or other variations, may be chosen.

SIMULATION RESULTS

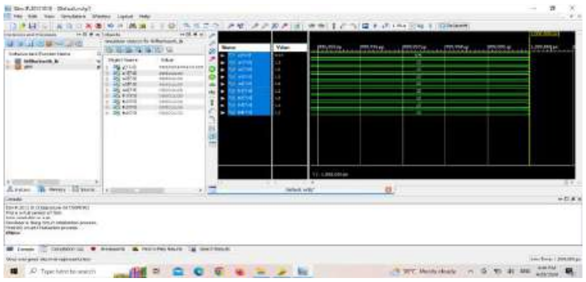

Fig 8.1 Simulation Result

BLOCK DIAGRAM

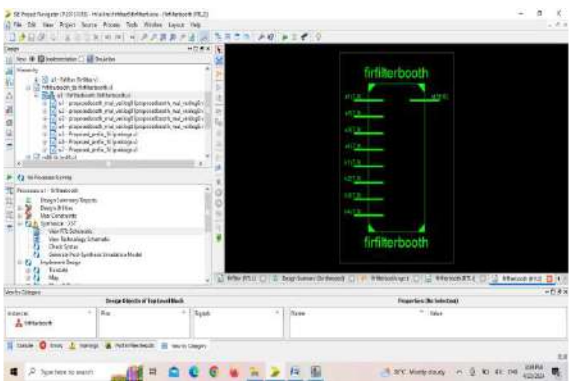

Fig 8.2 Block Diagram

RTL SCHEMATIC

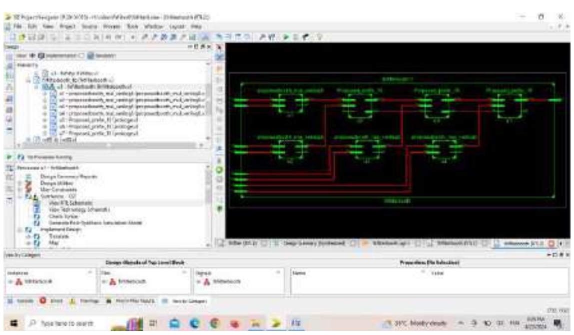

Fig 8.3 RTL Schematics

DEVICE UTILIZATION SUMMARY:

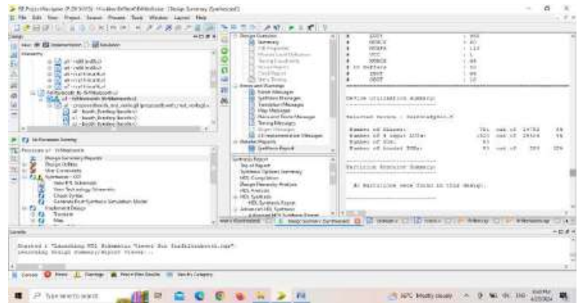

Fig 8.4 Area Report

TIMING SUMMARY

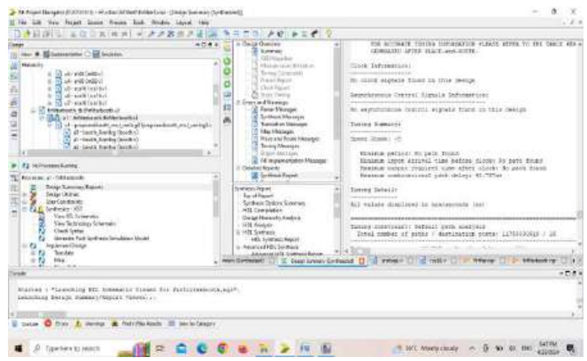

Fig 8.5 Delay Report

CONCLUSION

We have suggested pipeline FIR filters in this research that can achieve a very high throughput of more than 1:85 GSPS. A reference full-parallel architecture based on MBE, hierarchical Wallace tree network, and RCA is the first step in the suggested design. An effective pipelining technique at the gate level was established with the aid of a precise study that approximated the FIR filter's propagation delay in terms of unit gate delay. Because of this, the suggested full-parallel design may use fine-grained seamless pipelining to achieve extremely fast throughput. Alternative constructions that offer a comparatively high throughput while drastically lowering the area have also been presented in this article. This study is significant because the suggested FIR filter can scale

DSP applications that need a very high throughput rate more than giga samples per second.

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 nonrecursive 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] 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

[6] 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.

[7] 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.

[8] 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

[9] K Anjali Rao, Abhishek Kumar, Neetesh Purohit, "Efficient implementation for 3-parallel linear- phase FIR digital odd length filters " in proc. IEEE CICT, 2020

[10] A.Kumar, S. Yadav and N. Purohit, "Exploiting coefficient symmetry in conventional polyphase FIR filters," IEEE Access, vol.7, 2019

[11] 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.

[12] 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.

[13] D. K. Kahar and H. Mehta, "High-speed Vedic multiplier used Vedic mathematics, " in Proc. ICICCS, 2018.

[14] Rajesh K., Reddy G. "FPGA implementation of multiplier accumulator unit using Vedic multiplier and reversible gates " in proc. ICISC, 2019

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