

## **Digital Parallel FIR Filters using Different Multiplier and Adder: A Study**

**Sonam Shukla**

VLSI M. Tech. Scholar, Department of Electronics and Communication Engineering,  
Vidhyapeeth Institute of Science and Technology, Bhopal, M.P., India  
[sonam.shukla08@gmail.com](mailto:sonam.shukla08@gmail.com)

**Prof. Aditya Mishra**

Assistant Professor, Department of Electronics and Communication Engineering,  
Vidhyapeeth Institute of Science and Technology, Bhopal, M.P., India  
[adivist@gmail.com](mailto:adivist@gmail.com)

### **Abstract**

Finite Impulse Response (FIR) filters are essential components in digital signal processing (DSP) systems, offering inherent stability and linear phase characteristics. With growing demands for real-time, high-throughput signal processing, efficient hardware implementation of FIR filters has become increasingly critical. This study presents a comparative analysis of Digital Parallel FIR Filter architectures using different types of multipliers and adders to optimize speed, area, and power consumption. Various multipliers, such as Booth Multiplier, Wallace Tree Multiplier, and Array Multiplier, are examined in conjunction with fast adder circuits including Carry Look-Ahead Adder (CLA), Carry Select Adder (CSLA), and Common Boolean Logic (CBL) Adders. The proposed design methodology emphasizes parallel processing to enhance computational efficiency. Simulation results using hardware description languages (VHDL/Verilog) and synthesis on FPGA platforms demonstrate that certain combinations, such as the Booth Multiplier with CSLA, offer notable improvements in execution time and power efficiency. This study aims to guide VLSI designers in selecting optimal arithmetic units for FIR filter implementation based on application-specific constraints and performance requirements.

**Keywords:** - Digital Signal Processing, Adder, Multiplier, 2x2 Parallel FIR Filter

### **1. INTRODUCTION**

Finite Impulse Response (FIR) filters play a crucial role in various digital signal processing (DSP) applications, including audio processing, image enhancement, and communication systems. FIR filters are preferred due to their inherent stability and ability to achieve exact linear phase characteristics. However, with the increasing need for high-speed data processing in modern applications, the design and implementation of FIR filters demand efficiency in terms of area, power, and speed.

The architecture of an FIR filter typically involves a series of multipliers and adders, whose performance directly influences the filter's overall efficiency. In parallel FIR filter designs, the operations are executed concurrently, significantly improving processing speed and throughput. However, this enhancement often comes at the cost of increased hardware complexity and power consumption. Therefore, selecting efficient arithmetic units—specifically multipliers and adders—is crucial for optimizing the overall performance of FIR filters [1, 2].

This study focuses on analyzing and comparing FIR filters implemented using different types of multipliers and adders in a parallel configuration. The performance of traditional Array Multipliers, Wallace Tree Multipliers, and Booth Multipliers is investigated. Similarly, various adders such as Ripple Carry Adder (RCA), Carry Look-Ahead Adder (CLA), Carry Save Adder (CSA), and Common Boolean Logic (CBL) Adders are studied for their impact on speed and resource utilization.

The objective is to design parallel FIR filters using different combinations of these arithmetic units and evaluate them using VLSI simulation tools. The performance parameters considered include gate count, delay, area utilization, and power consumption. Field-Programmable Gate Arrays (FPGA) are used as the target implementation platform due to their flexibility and suitability for rapid prototyping [3].

By comparing these architectures, the study aims to identify the optimal combinations of multipliers and adders that yield high-performance, low-power FIR filters. The insights gained can significantly benefit digital hardware designers in making informed decisions during FIR filter implementation for real-time embedded DSP applications [4, 5].

On the other hand, two methods utilized in DSP applications that can be employed to lower power consumption are pipelining and parallel processing. Pipelining shortens the critical path by interleaving pipelining latches along the data path, at the price of increasing the number of latches and the system latency, whereas parallel processing enhance the sampling rate by

replicating circuitry, which requires more space, in order to process numerous inputs in parallel and generate multiple outputs simultaneously [8]. By decreasing the supply voltage, both methods can minimize power consumption without increasing sampling speed. The digital FIR filter's parallel processing will be covered in this study.

## **2. LITERATURE REVIEW**

**S Narendran et al. [1]**, Examining the FIR channel's display in light of the FFA calculation is one of the examination's main goals. The proposed structure focuses on region improved adders which are intended for symmetric coefficient that diminishes equipment cost in the plan with an oblige that the channel tap is of even request. Region investigation of silicon chip is performed utilizing various adders like Ripple Carry Adder (RCA), Carry Look-Ahead Adder (CLA) and Conditional Sum Adder (COS). The RCA uses less region than the other three adders, according to the execution evaluation report for a three equal 96-tap channel, which is based on the aforementioned viper. The symmetric convolution-based RCA Fast FIR Algorithm (FFA) structure requires just 896 cuts while COS requires 1024 cuts and CLA requires 960 cuts. For higher-request, FIR channel design of 1024 or higher taps, the reserve funds in the space usage for RCA based FFA is roughly 10% and 15% contrasted with CLA and COS based FFA separately in light of the fact that COS requires more equipment building blocks. The RCA based FFA expands the idleness by around 9% and 7% contrasted with CLA and COS based FFA separately, and there will be a compromise.

**K. Anjali Rao et al. [2]**, in this paper, we propose 3-equal limited motivation reaction (FIR) structures with the use of polyphase coefficient evenness and quick FIR calculations (FFAs). For symmetric convolution of odd length channel The proposed structures are advantageous as far as the quantity of multipliers which can lessen the execution intricacy of the equal FIR channels. FFAs are used to reduce the equal subfilter portions of a polyphase channel after first reconfiguring 3-equal conventional FIR structures to allow the polyphase coefficient evenness of odd length FIR channels to be applied for the reduction of the half number of multipliers. The suggested FFA-based structure is unmatched in terms of the reduced number of multipliers needed when comparing the presentation with current comparison designs.

**Yi Zheng et al. [3]**, to make students ace an equipment execution of computerized channels, we take the plan of an equal FIR advanced low-pass channel as a showing case, which can introduce the execution strategy for computerized channels joining Matlab and an equipment

portrayal language together. In the first place, as per the undertaking of the showing case, the coefficients of FIR advanced not entirely settled by a FDATool utilization of Matlab. By standardization and quantization, the channel coefficients and examined information are both changed over from the drifting point information to the fixed-point information. The digit width of substantial information pieces of FIR channel still up in the air as per the standard of increase gather activity of FPGAs. Then, at that point, we use Verilog HDL to execute a quick multiplier and the equal FIR computerized low-pass channel. At long last, we contrast the channel results of FPGAs and the ones of Matlab, and investigate the explanation. The case showing technique, which consolidates Matlab and FPGAs, is useful to foster the equipment execution capacity of computerized signal handling of students.

**S. Sreekanth et al. [4]**, in this paper, Polyphase Finite Impulse Response (FIR) computerized channel is planned and carried out by utilizing 40 nm LP CMOS TSMC Library. For some advanced channelized collectors and transmitters in computerized correspondence frameworks the speed of FIR channel and idleness are significant elements. Polyphase channel is executed with equal methodology in which equal multipliers, adders and FFT block utilizing the Radix-2 calculation are utilized. Radix-2 purposes not many multipliers contrasted with Radix-2 calculation. In this work elite execution is accomplished contrasted with ordinary direct structure plan and different plans.

**Qiaoyu Tian et al. [5]**, the Cook-Toom calculation is generally utilized in short-length direct convolution, which is the structure square of enormous focuses convolution calculations. This paper proposes further developed equal limited motivation reaction (FIR) channel structures for direct stage FIR channel, which depends on the Cook-Toom calculation. In the proposed structures, Cook-Toom calculation is utilized to diminish the quantity of sub-channels, and the symmetric properties of the straight stage FIR channel's coefficients is utilized to additionally lessen the quantity of multipliers in sub-channels. Contrasted and the announced FFA and ISCA equal FIR channel structures, the proposed technique can significantly decrease the computational intricacy. In particular, for a 8-equal 144-tap channel, the proposed plan saves 18 multipliers (5%), 45 adders (7.9%) contrasted and the construction in light of Winograd convolution calculation.

**Payal Paliwal et al. [6]**, equal FIR channel is the need of many low power and rapid DSP applications. In this paper, quick FIR calculation based equal symmetric FIR channel utilizing Han-Carlson snake based vedic multiplier is proposed. FFA calculation lessens the multiplier

consider contrasted with the conventional equal plan. To work on the presentation of the proposed channel, as of late evolved Han-Carlson viper based Vedic multiplier is utilized. In the proposed plan the viper unit is additionally carried out utilizing Han-Carlson snake. In the proposed plan two and three equal FIR channels of request 24 and 72 are executed utilizing VHDL. The execution results show that the proposed engineering gives low basic way deferral and power dispersal when contrasted with the regular one. With the upside of low postponement and power, proposed design is valuable in present day signal handling and correspondence applications.

**Swetha Annangi et al. [7]**, in this paper, a 144-tap 16-equal Fast Finite Impulse Response (FIR) Algorithm (FFA) channel structure is planned utilizing verilog HDL. The planned channel structure is mimicked utilizing XILINX ISE 14.7. The planned module is combined utilizing CADENCE RTL Compiler and the application explicit incorporated circuit (ASIC) plan of the proposed channel structure is executed utilizing CADENCE apparatus set on CADENCE GPDK45nm innovation. By applying Fast FIR Algorithm, 65% of multipliers are diminished and the quantity of adders are expanded. The adders possess less silicon region than multipliers. Subsequently, the decrease in region is accomplished by supplanting the multipliers with adders. Further, the proposed 16 equal FFA channel structure lessens the postpone when contrasted with 16 equal FIR channel structure without applying Fast FIR Algorithm. The proposed channel engineering involves an area of 79220 sq.  $\mu\text{m}$  and consumes a force of 20mW at 333MHz when integrated.

**Shalina Percy et al. [8]**, have the requirement for productive Finite Impulse Response (FIR) channels in fast applications targets Field Programmable Gate Arrays (FPGAs) as a powerful and adaptable stage for computerized execution. In spite of the fact that FIR channel offer benefits like direct stage trademark, no criticism circles and great framework strength, its convolution nature balances a test in parallelization because of information reliance and computational intricacy. To determine this, we propose a clever FPGA-based reconfigurable channel engineering, which processes a few information tests in equal and separates information interdependency in a twisting design. This conventional pipelined-equal channel is parameterizable as far as channel request and level of parallelization. Exploratory outcomes show a throughput of 7.2 GSPS with a working recurrence of just 450 MHz for a channel length of 11 with 16 equal sources of info. With parallelization of 4, it is 4.64 times quicker than the

best in class answer for a channel length of 16 and a promising 41% expansion in throughput is accomplished for a higher request of 61.

### 3. 2×2 PARALLEL FIR FILTER

In general, two parallel FIR filter can be expressed as traditional two parallel digital filters is shown in figure 1. For this two parallel FIR filter  $L=2$ . This will require three FIR sub-filter blocks of length  $N/2$ , one pre-processing adder and three post-processing adders. Total number of multiplier and adders required are  $3N/2$  and  $3(N/2-1) + 4$  respectively.

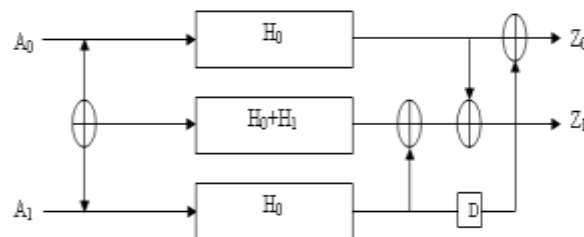


Figure 1: Parallel 2×2 FIR Filter

Following are the equations used to design the two parallel FIR filter with two inputs  $A_0$ ,  $A_1$  and two outputs  $Z_0$ ,  $Z_1$ . For implementing this filter three FIR sub-filter blocks has been used as compare to traditional two FIRs sub-block filter, having length  $N/3$ . Two of three sub-filters  $H_0+H_1$  and  $H_0-H_1$  are having symmetric coefficient which reduces the number of multiplier and adders. Here two preprocessing and four post-processing adders have been used along with delay equipment. The symmetric sub-filter block has been implemented at the cost of two additional adders among those one is pre-processing and other one is post-processing for  $L=2$ . Following are the equations used to design the filter:

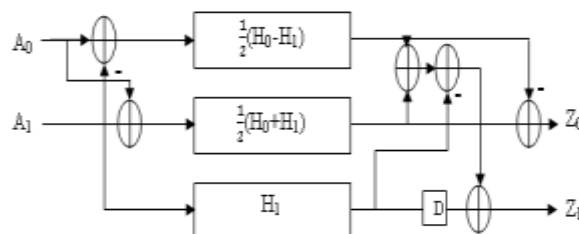


Figure 2: Proposed Parallel 2×2 FIR Filter

This same process is used for the  $n$  number of bits and thus we get the final sum and carry as output.

Example 1: Consider a 33-tap FIR filter with a set of symmetric coefficient as follows:

$\{h(0), h(1), h(2), h(3), h(4), \dots, h(29), h(30), h(31)\}$



Where

$$h(0) = h(32),$$

$$h(1) = h(31),$$

$$h(2) = h(29),$$

$$h(3) = h(28)$$

.....

$$h(12) = h(20)$$

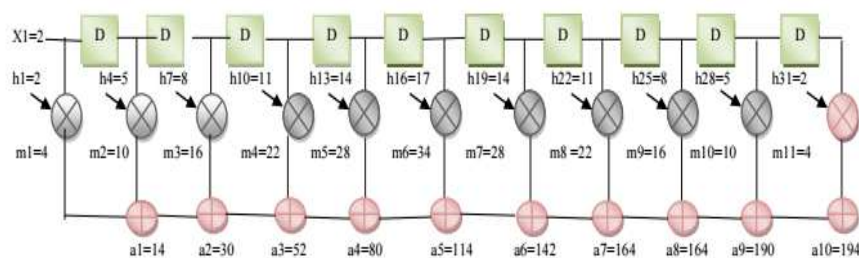


Figure 3: Internal Structure of H1

The symmetric parallel FIR filter is shown in Figure 3. The three parallel FIR filter consists of filter blocks. The input to the system is represented as  $A_0$ ,  $A_1$  and the response of the system as  $Z_0$  and  $Z_1$ . Let  $X_0=5$ ,  $X_1=2$ ,  $X_2=3$ . The filter blocks H1 with its mod 3 coefficients are shown in Figure 3.

#### 4. ADDER AND MULTIPLIER

##### Parallel Adder:-

Parallel adder can add all bits in parallel manner i.e. simultaneously hence increased the addition speed. In this adder multiple full adders are used to add the two corresponding bits of two binary numbers and carry bit of the previous adder. It produces sum bits and carry bit for the next stage adder. In this adder multiple carry produced by multiple adders are rippled, i.e. carry bit produced from an adder works as one of the input for the adder in its succeeding stage. Hence sometimes it is also known as Ripple Carry Adder (RCA). Generalized diagram of parallel adder is shown in figure 4.

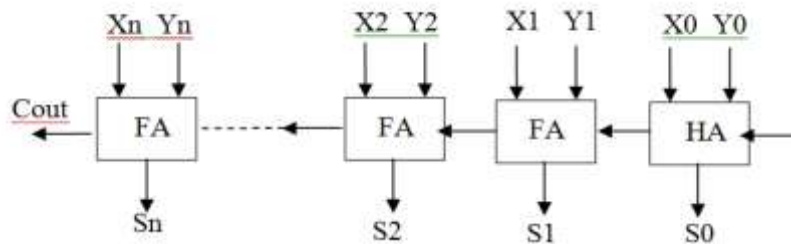


Figure 4: Parallel Adder ( $n=7$  for SPFP and  $n=10$  for DPFP)

### Carry Skip Adder:-

This adder gives the advantage of less delay over Ripple carry adder. It uses the logic of carry skip, i.e. any desired carry can skip any number of adder stages. Here carry skip logic circuitry uses two gates namely “and gate” and “or gate”. Due to this fact that carry need not to ripple through each stage. It gives improved delay parameter. It is also known as Carry bypass adder. Generalized figure of Carry Skip Adder is shown in figure 5.

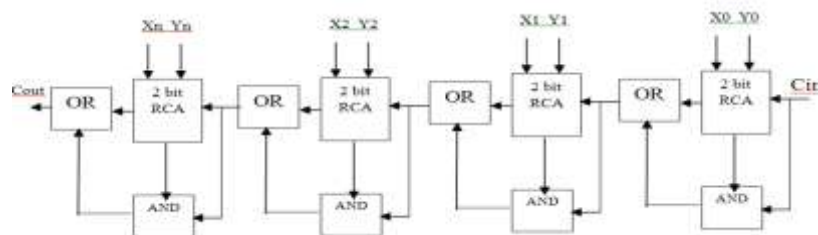


Figure 5: Carry Skip Adder

### Carry Select Adder:-

Carry select adder uses multiplexer along with RCAs in which the carry is used as a select input to choose the correct output sum bits as well as carry bit. Due to this, it is called Carry select adder. In this adder two RCAs are used to calculate the sum bits simultaneously for the same bits assuming two different carry inputs i.e. ‘1’ and ‘0’. It is the responsibility of multiplexer to choose correct output bits out of the two, once the correct carry input is known to it. Multiplexer delay is included in this adder. Generalized figure of Carry select adder is shown in figure 6. Adders are the basic building blocks of most of the ALUs (Arithmetic logic units) used in Digital signal processing and various other applications. Many types of adders are available in today’s scenario and many more are developing day by day. Half adder and Full adder are the two basic types of adders. Almost all other adders are made with the different arrangements of these two basic adders only. Half adder is used to add two bits and produce



sum and carry bits whereas full adder can add three bits simultaneously and produces sum and carry bits.

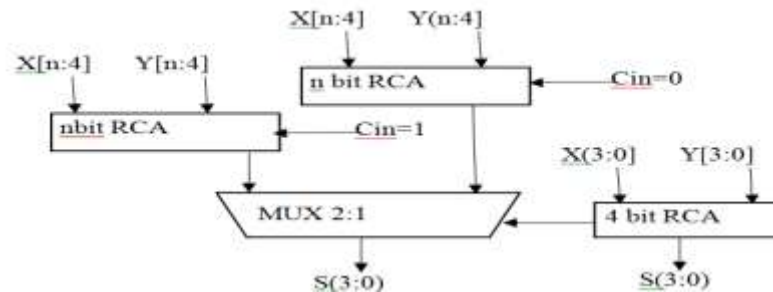


Figure 6: Carry Select Adder

### Multiplier:-

All the parallel, the pipeline and the high performance multipliers implemented in hardware is illustrated in the block diagram of Figure 7. It consists of three phases as follows.

- Partial Product Generator (PPG)
- Partial Product Reduction Tree (PPRT)
- Carry Propagate adder (CPA)

The PPG creates fractional items either utilizing two info OR door or by utilizing MBE strategy. The PPRT is utilized to pack the few varieties of halfway item pushes into a variety of two lines. The CSA and different kinds of blowers can be utilized as the PPRT. At last the CPA is utilized for the expansion of a variety of two lines to get the result of augmentation. The fastest CPA is the CLA).

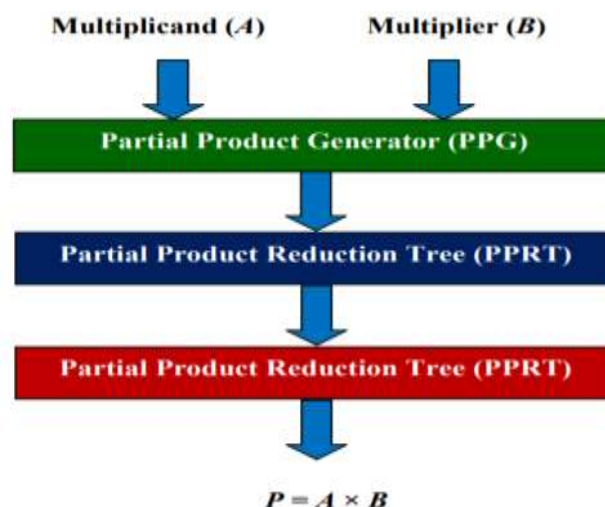


Figure 7: Block diagram of Multiplier

The non-pipeline multiplier can be changed over into the pipeline multiplier by embedding register between the phases of combinational rationale circuits.

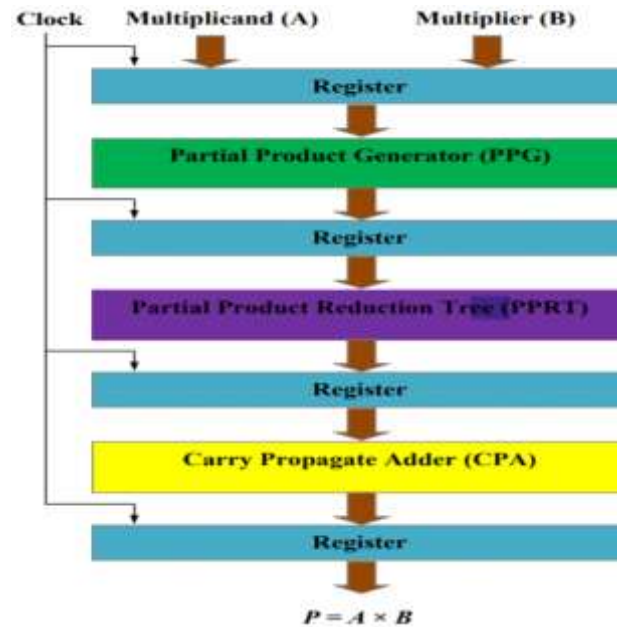


Figure 8: Block diagram of Pipeline Multiplier

Figure 8 shows the square outline of the pipeline multiplier. All the pipeline stages are worked with the synchronous clock signal. Each clock cycle, two new operands ( $A_n$ ) and ( $B$ ) is gotten by the pipeline and handled in the progressive phases of the pipeline. When the pipelines arrange is topped off, at that point each clock cycle another result of duplication activity is produced.

## 5. CONCLUSION

This study presents a detailed analysis and comparison of digital parallel FIR filter architectures implemented using various combinations of multipliers and adders. FIR filters are fundamental in digital signal processing applications, and their performance significantly depends on the efficiency of the underlying arithmetic units. Through this study, multiple combinations were explored—such as Array Multiplier with Ripple Carry Adder, Wallace Tree Multiplier with Carry Look-Ahead Adder, and Booth Multiplier with Common Boolean Logic Adder—to determine the optimal design for speed, power consumption, and area utilization.

The simulation and synthesis results demonstrated that Booth Multipliers combined with advanced adders like the Carry Select Adder (CSLA) or Common Boolean Logic (CBL) Adders offered substantial improvements in execution speed and resource optimization compared to traditional designs. Wallace Tree Multipliers also showed competitive performance in terms of speed due to their hierarchical reduction structure, although they can be more complex to implement.

Moreover, parallel FIR filter designs significantly enhanced throughput but introduced challenges related to power and area, which were mitigated through the choice of efficient arithmetic circuits. It is concluded that the selection of appropriate multiplier and adder combinations plays a critical role in determining the overall efficiency of FIR filter architectures, especially for real-time DSP applications.

Future work can explore adaptive FIR filters, low-power approximation techniques, and the application of machine learning-based optimization algorithms to further improve the performance of these digital filter designs.

## REFERENCES

- [1] S Narendran and B T Geetha, “Performance Analysis of Parallel FIR Digital Filter Based on Even Symmetric Fast FIR Algorithm using Different Adders”, 5th International Conference on Electronics, Communication and Aerospace Technology (ICECA), IEEE 2021.
- [2] K. Anjali Rao;Abhishek Kumar;Neetesh Purohit, “Efficient Implementation for 3-Parallel Linear-Phase FIR Digital Odd Length Filters”, IEEE 4th Conference on Information & Communication Technology (CICT), IEEE 2020.
- [3] Yi Zheng;Ping Zheng, “Case Teaching of Parallel FIR Digital Filter Design Combined Matlab with FPGAs”, International Conference on Artificial Intelligence and Education (ICAIE), IEEE 2020.
- [4] S. Sreekanth; Pratima B. Shinde; G. Vijaya Durga, “Performance Analysis of Higher Order FIR Polyphase Filter”, Second International Conference on Intelligent Computing and Control Systems (ICICCS), IEEE 2018.
- [5] Qiaoyu Tian;Yinan Wang;Guiqing Liu;Xiangyu Liu;Jietao Diao, Hui Xu, “Hardware Efficient Parallel FIR Filter Structure Based on Modified Cook-Toom Algorithm”, IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), IEEE 2018.

- [6] Payal Paliwal;Janki Ballabh Sharma, “Efficient FPGA Implementation Architecture of Fast FIR Algorithm Using Han-Carlson Adder Based Vedic Multiplier”, International Conference on Inventive Research in Computing Applications (ICIRCA), IEEE 2018.
- [7] Swetha Annangi;Ravisankar Puli, “ASIC implementation of efficient 16-parallel fast FIR algorithm filter structure”, 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), IEEE 2017.
- [8] Shalina Percy Delicia Figuli;Peter Figuli;Jürgen Becker, “A reconfigurable high-speed spiral FIR filter architecture”, 40th International Conference on Telecommunications and Signal Processing (TSP), IEEE 2017.
- [9] Shahnam Mirzaei, Anup Hosangadi, Ryan Kastner, “FPGA Implementation of High Speed FIR Filters Using Add and Shift Method”, IEEE 2006.
- [10] Amina Naaz.S, Mr.Pradeep M.N, Satish Bhairannawar and Srinivas halvi, “FPGA Implementation Of High Speed Vedic Multiplier using CSLA For Parallel Fir Architecture”, 2014 2nd International Conference on Devices, Circuits and Systems (ICDCS).
- [11] Laxman P.Thakre, Suresh Balpande, Umesh Akare, Sudhir Lande, “Performance Evaluation and Synthesis of Multiplier used in FFT operation using Conventional and Vedic algorithms,” Third international conference on emerging trends in Engineering and Technology , IEEE, 2010.
- [12] S. S. Kerur, Prakash Narchi, Jayashree C N, Harish M Kittur and Girish V. A., “Implementation of Vedic Multiplier for Digital Signal Processing,” International Conference on VLSI ,Communication & Instrumentation (ICVCI), 2011.
- [13] G.Vaithiyanathan, K.Venkatesan, S.Sivaramakrishnan, S.Sivaand, S.Jayakumar, “Simulation and implementation of Vedic multiplier usingVHDL code,” International Journal of Scientific & Engineering Research, vol.4, 2013.
- [14] Pushpalata Verma and K. K. Mehta, “Implementation of an Efficient Multiplier based on Vedic Mathematics Using EDA Tool,” International Journal of Engineering and Advanced Technolog(IJEAT), vol.1, June 2012.
- [15] C. Cheng and K. K. Parhi, “Furthur complexity reduction of parallel FIR filters,” in Proc. IEEE ISCAS, May 2005, vol. 2, pp. 1835–1838.
- [16] C. Cheng and K. K. Parhi, “Low-cost parallel FIR structures with 2-stage parallelism,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 54, no. 2, pp. 280–290, Feb. 2007.



- [17] J. G. Chung and K. K. Parhi, “Frequency-spectrum-based low-area low-power parallel FIR filter design,” EURASIP J. Appl. Signal Process., vol. 2002, no. 9, pp. 444–453, Jan. 2002.
- [18] K. K. Parhi, VLSI Digital Signal Processing systems: Design and Implementation. New York: Wiley, 1999.
- [19] Nivedita A. Pande, Vaishali Niranjane, Anagha V. Choudhari, “ Vedic Mathematics for Fast Multiplication in DSP,” International Journal of Engineering and Innovative Technology (IJEIT) ,vol.2, 2013.