

VLSI Architecture for FIR Filter using Radix-4 Booth Multiplier and CBL Adder

Teepu Sultan

M. Tech. Scholar, Department of Electronics and Communication, Bhabha Engineering Research Institute, Bhopal

Prof. Suresh S. Gawande

Guide, Department of Electronics and Communication, Bhabha Engineering Research Institute, Bhopal

Abstract

The main objective of this research paper is to design architecture for finite impulse response (FIR) filter using radix-4 booth multiplier and common Boolean logic (CBL) adder. Finite Impulse response (FIR) filters are extensively utilized in digital signal processing in which different filter parts operate at different rates. It has applications in communication transmitters and receivers. FIR filters when implemented use multipliers and accumulators. There are various types of multiplier structure algorithms and their variations such as Combinational multiplier, Wallace Tree multiplier, Array multiplier and Sequential multiplier and Booth multiplier. Booth multipliers reduce the resulting number of partial products generated as a result of multiplication of two binary numbers. This paper presents a VLSI architecture for FIR filters integrating the Radix-4 Booth Multiplier and Carry Bypass Logic (CBL) Adder, improving speed, power efficiency, and area utilization. Performance comparisons with conventional architectures highlight the benefits of the proposed design.

Keywords – Common Boolean Logic Adder, Xilinx Software, Finite Impulse Response, Radix-4, Booth Multiplier

I. INTRODUCTION

Since it settles to zero in a certain amount of time, a finite impulse response (FIR) channel in signal processing is one whose response to any input of limited length has a limited term. As opposed to infinite impulse response (IIR) channels, which may remain inconclusive (mostly declining) and may contain inward input. Numerous DSP applications make extensive use of FIR channels. The FIR channel circuit must be able to operate at high example rates in certain applications and low-control circuits that operate at direct example rates in other applications. Computerized FIR channels can be coupled with parallel (or square) handling to increase the overall throughput or reduce the power consumption of the first channel [1, 2].

Although the use of sequential FIR channels has been widely considered, very little work has been done to directly reduce the equipment's multi-sided quality or power consumption of parallel FIR channels. The equipment components that are present in the first channel are typically replicated when parallel processing is applied to a FIR channel. The multiplier circuit is topology also affects the power consumption that results. Picking multipliers with more equipment expansiveness as opposed to profundity would decrease the postponement, as well as the aggregate power utilization. A considerable measure of outline strategies for low power computerized FIR channel have been proposed, for instance, a strategy executing FIR channel utilizing simply enrolled adders and hardwired shifts exist [3, 4].

Parallel duplication is utilized to meet out the present prerequisite. Two kinds of parallel augmentations are exhibit duplication and tree increase. The fundamental multiplier is a basic cluster multiplier and it is planned in view of move and – include task. One of the cases for exhibit increase is the Braun multiplier and is intended for unsigned paired numbers. For tree structure Wallace multiplier is outlined and it is likewise for an unsigned double numbers. In the exhibit augmentation, for marked numbers Baugh – Wooley, Booth Multiplier and Modified Booth Algorithm (MBA) are utilized. Dadda is another kind of multiplier in light of tree structure and is utilized for the increase of the marked numbers. These traditional double multipliers for unsigned numbers are considered for examination. Vedic arithmetic is the arrangement of science followed in old India and mostly manages Vedic scientific formulae and their applications to different branches of math. The word 'Vedic' is gotten from the word 'Veda' which implies the storage facility of all information [5, 6].

After studying the Vedas for eight years, Sri Bharati Krishna Tirthaji (1884–1960) recreated Vedic science from the ancient Indian sacred texts. According to his analysis, thirteen sub-end products known as Sutras and sixteen standards or word-formulae serve as the main foundation for Vedic mathematics. This topic is really fascinating and has several useful computations that relate to other engineering specialties, such computing and digital signal processing. The many-sided nature of the figures found in conventional arithmetic is diminished by Vedic knowledge. In Vedic arithmetic, sixteen sutras are often available [7].

Of these, only two sutras are relevant to heightened activity. They are Nikhilam Sutra (which actually means "all" from 9 and "last" from 10) and Urdhava Triyakbhyam Sutra (which actually means "vertically and across"). Urdhava-Triyakbhyam is a general augmentation method. The Urdhava Triyakbhyam sutra's reasoning is particularly similar to that of the traditional cluster multiplier. Here, a similar reasoning used for decimal numbers is used to find the paired usage of this calculation. The double usage of Nikhilam Sutra isn't yet effective [8, 9]. In this research paper, a novel architecture of FIR filters is based on radix-4 booth multiplier and common Boolean logic adder.

II. PROPOSED METHODOLOGY

A FIR filter, also known as a recursive filter, employs past output values in addition to input values. These are stored in the processor's memory, just as the previous information. In reality, the term "recursive" means "running back" and refers to the way that previously calculated yield values backpedaling into the count of the most current yield. This articulation for a recursive channel includes phrases in y_n , y_{n-1} , y_{n-2} , and other terms in addition to the info esteems (x_n , x_{n-1} , x_{n-2} , etc.).

According to this explanation, FIR channels need to have more counts done since they have input terms in addition to prior yield terms in the channel articulation. Compared to a proportionate non-recursive channel, a recursive channel typically requires a significantly lower arrange channel to complete a certain recurrence reaction characteristic, meaning that the processor must evaluate fewer words. The fact that these channels operate on discrete-time signals gives rise to the term "computerized channel" [4]. The term finite impulse response arises because the filter output is computed as a weighted, finite term sum, of past, present, and perhaps future values of the filter input, i.e.,

$$y[n] = \sum_{k=-M_1}^{M_2} b_k x[n - k] \quad (1)$$

Where both M_1 and M_2 are finite

One of the simplest FIR filters that may be considered is a 3-term moving average filter of the form

$$y[n] = \frac{1}{3}(x[n + 1] + x[n] + x[n - 1]) \quad (2)$$

An FIR filter is based on a feed-forward difference equation– Feed-forward means that there is no feedback of past or future outputs to form the present output, just input related terms [5].

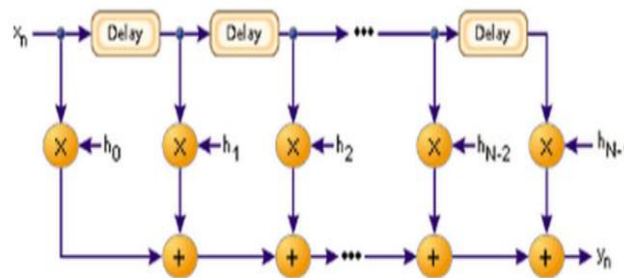


Figure 1: Logical Structure of FIR Filter

CBL adder:

Area and power efficient excessive speed facts logic path are the most enormous regions of studies. With the help of simple change in gate level we will obtain the development inside the effects. velocity of the adder depends on the time required to propagate the bring thru the adder. those adder works in series layout, this is the sum of the primary position bit is calculated while the preceding bits are summed and the convey is propagated to that subsequent level.

Carry select adder (CSLA) is one of the superior adders used in information processing processors to perform fast arithmetic function. It specializes in the hassle of bring propagation put off through producing the deliver independently at each degree and the pick out the efficient one with the assist of multiplexer to perform the sum. The traditional CLSA is RCA (Ripple

carry adder) which generate the partial sum and carry by way of the use of the enter deliver circumstance $C_{in}=0$ and $C_{in}=1$, select one out of each pair to shape final sum and final convey output.

RCA isn't location efficient as huge wide variety of gates circuitry is used to form the partial merchandise after which the final sum and convey is selected.

Another shape of CLSA adder makes use of binary to excess-1 convertor changing ripple deliver adder with $C_{in}=1$. This adder is known as CLSA at the side of BEC. The range of gates used has been reduced while we must layout big bit adder. This adders is more conventional as examine to RCA while cope with silicon vicinity used however that is having marginally higher put off time.

The proposed not unusual Boolean logic (CBL) adder is place-power-put off efficient. It paintings on the good judgment to get rid of the redundant adders and use commonplace Boolean common sense as examine to standard deliver pick adder.

The CBL block is constructed from two components sum technology block and carry era block. In sum generation block the output sum is completed using the multiplex. This multiplex is used to choose the output cost depeding at the value of C_{in} (previous bit).

If $C_{in}=0$, then output is xor of the two enter bits. If $C_{in}=1$, then output get inverted. In deliver generation block, multiplexer is used to pick out the delivery of next degree relying upon the previous carry enter. If $C_{in}=0$, cout is OR of two input and if $C_{in}=1$ the output deliver is

AND of the input bit.

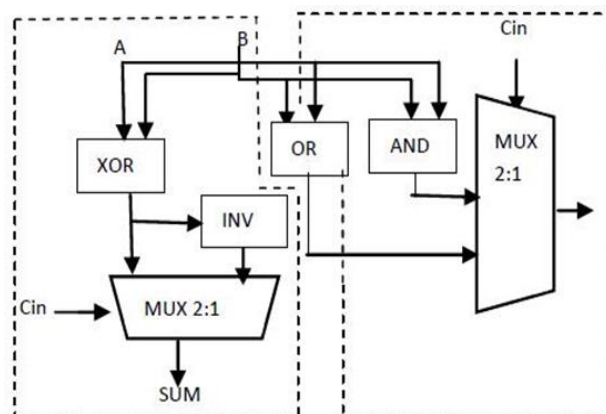


Figure 2: Block Diagram of CBL

$If C_{in} = 0$
 $Sum = A XOR B$
 $Carry = A OR B$
 else
 $Sum = NOT (A XOR B)$
 $Carry = A AND B$

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

III. RADIX-4 ALGORITHM

To further decrease the number of partial products, algorithms with higher radix value are used. In radix-4 algorithm grouping of multiplier bits is done in such a way that each group consists of 3 bits as mentioned in table 1. Similarly the next pair is the overlapping of the first pair in which MSB of the first pair will be the LSB of the second pair and other two bits. Number of groups formed is dependent on number of multiplier bits. By applying this algorithm, the number of partial product rows to be accumulated is reduced from n in radix-2 algorithm to n/2 in radix-4 algorithm. The grouping of multiplier bits for 8-bit of multiplication is shown in figure 3.

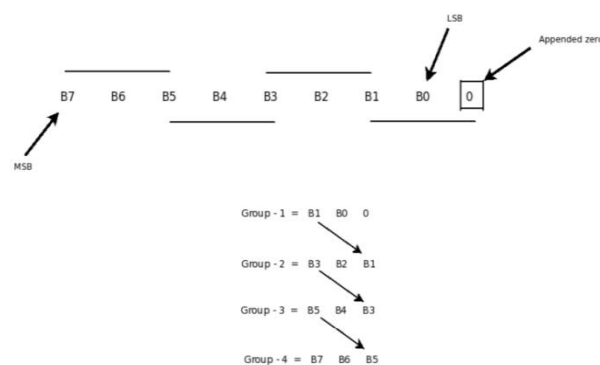


Figure 3: Grouping of multiplier bits in Radix-4 Booth algorithm

For 8-bit multiplier the number groups formed is four using radix-4 booth algorithm. Compared to radix-2 booth algorithm the number of partial products obtained in radix-4 booth algorithm is half because for 8-bit multiplier radix-2 algorithm produces eight partial products. The truth

table and the respective operation is depicted in table 1. Similarly when radix-8 booth algorithm is applied to multiplier of 8-bits each group will consists of four bits and the number of groups formed is 3. For 8x8 multiplications, radix-4 uses four stages to compute the final product and radix-8 booth algorithm uses three stages to compute the product. In this thesis, radix-4 booth algorithm is used for 8x8 multiplications because number components used in radix-4 encoding style.

Table 1: Truth Table for Radix-4 Booth algorithm

B_{i+1}	B_i	B_{i-1}	Operation	Y_{i+1}	Y_i	Y_{i-1}
0	0	0	+0	0	0	0
0	0	1	+A	0	1	0
0	1	0	+A	0	1	0
0	1	1	+2A	0	0	1
1	0	0	-2A	1	0	1
1	0	1	-A	1	1	0
1	1	0	-A	1	1	0
1	1	1	-0	1	0	0

IV. SIMULATION ANALYSIS

Simulation of these tests should be possible by utilizing Xilinx 14.2 I VHDL instrument. In this paper we are concentrating on engendering delay. Spread postpone must be less for better execution of advanced circuit.

As appeared in table I the quantity of cut, number of LUTs, delay are acquired for the complex Vedic multiplier utilizing basic Boolean rationale viper and past calculation. From the investigation of the outcomes, it is discovered that the complex Vedic multiplier utilizing basic Boolean rationale snake gives a predominant execution as contrasted and past calculation for Xilinx programming.

Take a look at the VTS and RTL of FIR_4tap in fig. 4 & fig. 5. In fig. 6, there is a simulation result for and in fig. 67there is a waveform.

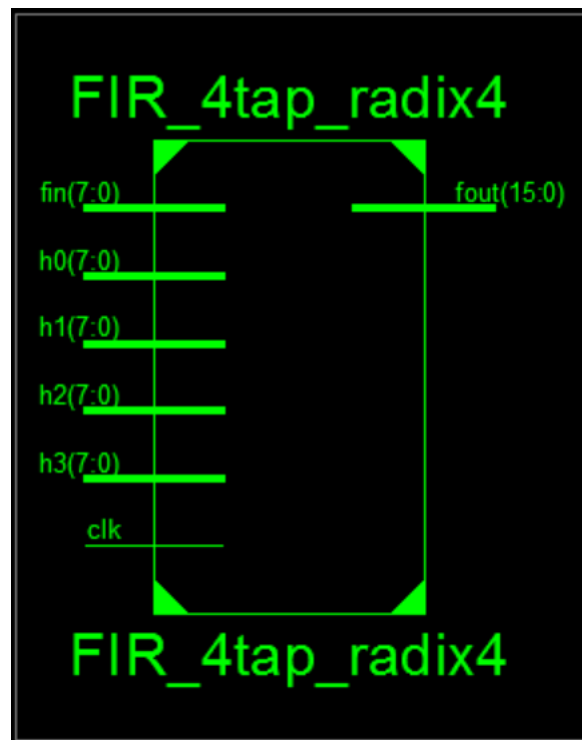


Figure 4: Examine VTS of FIR_4tap

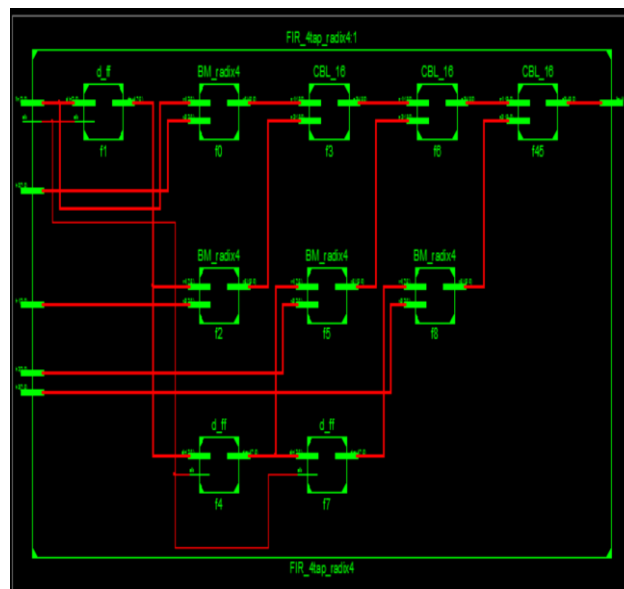


Figure 5: Examine RTL of FIR_4tap

Device utilization summary:

Selected Device : 6slx4tqg144-3

Slice Logic Utilization:

Number of Slice Registers:	28	out of	4800	0%
Number of Slice LUTs:	607	out of	2400	25%
Number used as Logic:	603	out of	2400	25%
Number used as Memory:	4	out of	1200	0%
Number used as SRL:	4			

Slice Logic Distribution:

Number of LUT Flip Flop pairs used:	622			
Number with an unused Flip Flop:	594	out of	622	95%
Number with an unused LUT:	15	out of	622	2%
Number of fully used LUT-FF pairs:	13	out of	622	2%
Number of unique control sets:	2			

IO Utilization:

Number of IOs:	57			
Number of bonded IOBs:	57	out of	102	55%

Timing Summary:

Speed Grade: -3

Minimum period: 1.662ns (Maximum Frequency: 601.811MHz)
Minimum input arrival time before clock: 2.644ns
Maximum output required time after clock: 22.522ns
Maximum combinational path delay: 23.081ns

Figure 6: Examine Simulation of FIR_4tap

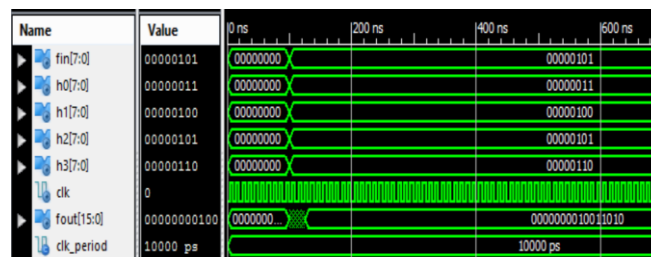


Figure 7: Examine Waveform of FIR_4tap

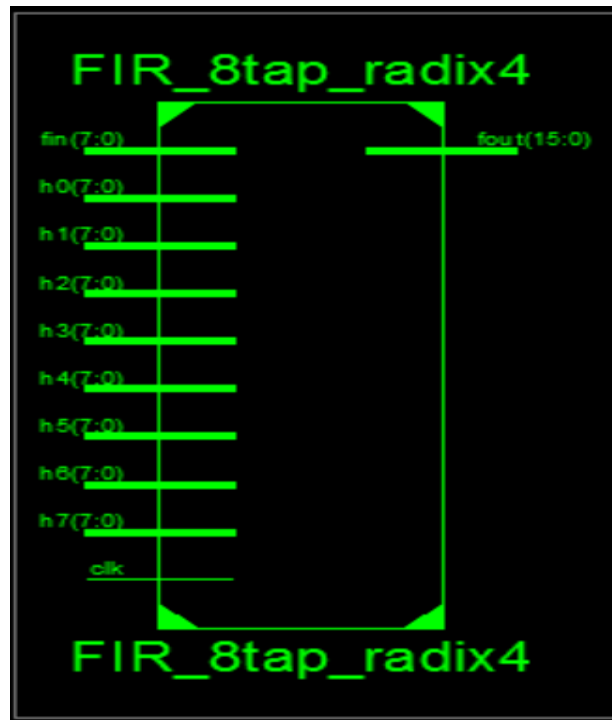


Figure 8: Examine VTS of FIR_8tap

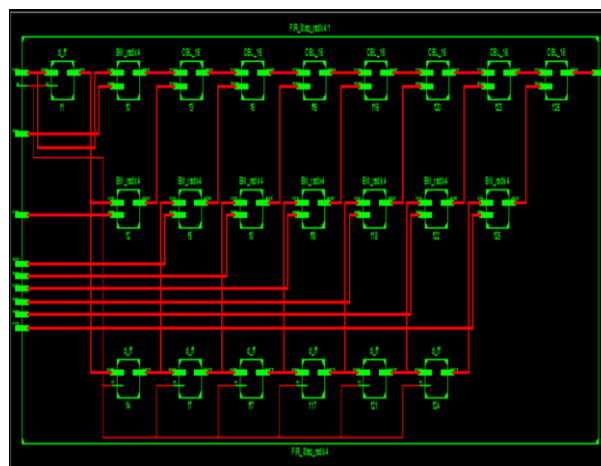


Figure 9: Examine RTL of FIR_8tap

```

Device utilization summary:
-----
Selected Device : 6slx4tqg144-3

Slice Logic Utilization:
Number of Slice Registers:          78 out of 4800    1%
Number of Slice LUTs:              1232 out of 2400    51%
    Number used as Logic:          1226 out of 2400    51%
    Number used as Memory:         6 out of 1200     0%
    Number used as SRL:            6 out of 1200     0%

Slice Logic Distribution:
Number of LUT Flip Flop pairs used: 1279
Number with an unused Flip Flop:    1201 out of 1279    93%
Number with an unused LUT:          47 out of 1279     3%
Number of fully used LUT-FF pairs:   31 out of 1279     2%
Number of unique control sets:       2

IO Utilization:
Number of IOs:                      89
Number of bonded IOBs:              89 out of 102    87%

Timing Summary:
-----
Speed Grade: -3

Minimum period: 1.682ns (Maximum Frequency: 594.513MHz)
Minimum input arrival time before clock: 2.644ns
Maximum output required time after clock: 28.615ns
Maximum combinational path delay: 29.515ns

```

Figure 10: Examine Simulation of FIR_8tap

V. CONCLUSION

The integration of the Radix-4 Booth Multiplier and CBL Adder in FIR filter architecture significantly enhances performance in VLSI implementations. The proposed design achieves lower delay, reduced power consumption, and optimized area utilization, making it ideal for high-speed digital signal processing applications.

REFERENCES

- [1] K. Sravani, M. Saisri, U. Vidya Sivani and A.Ramesh Kumar, "Design and Implementation of Optimized FIR Filter using CSA and Booth Multiplier for High Speed Signal Processing", 4th International Conference for Emerging Technology (INCET), IEEE 2023.
- [2] A. S. Kumar et al., "An Efficient AVR interfaced Bluetooth controlled Robotic Car system," 2023 13th International Conference on Cloud Computing, Data Science & Engineering (Confluence), India, pp. 499-502, 2023.

- [3] A. S. Kumar et al., "A Novel RRAM-based FPGA architecture with Improved Performance and Optimization Parameters," 2022 IEEE 19th India Council International Conference (INDICON), Kochi, India, pp. 1-5, 2022.
- [4] B. N. K. Reddy and A. S. Kumar, "An Efficient Low-Power VIP based VC Router Architecture for Mesh-based NoC," 2022 IEEE 19th India Council International Conference (INDICON), India, pp. 1-5, 2022.
- [5] Raghava Rao, K., Naresh Kumar Reddy, B. & Kumar, A.S. "Using advanced distributed energy efficient clustering increasing the network lifetime in wireless sensor networks", Soft Computing, 2023.
- [6] G. Shanthi, A. S. Kumar, et al., "An Efficient FPGA Implementation of Cascade Integrator Comb Filter," 2022 International Conference on Intelligent Innovations in Engineering and Technology (ICIET), pp. 151-156, 2022.
- [7] Sai Kumar, U. Siddhesh, N. Sai kiran and K. Bhavitha, "Design of High Speed 8-bit Vedic Multiplier using Brent Kung Adders," 2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT), pp. 1-5, 2022.
- [8] Sayed, J. F., Hasan, B. H., Muntasir, B., Hasan, M., & Arifin, F. (2021). "Design and Evaluation of a FIR Filter Using Hybrid Adders and Vedic Multipliers". 2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST).
- [9] S. China Venkateshwarlu, Mohammad, Chandra Shaker Pittala and Rajeev Ratna Vallabhuni , "Optimized Design of Power Efficient FIR Filter Using Modified Booth Multiplier", 4th International Conference on Recent Trends in Computer Science and Technology, IEEE 2021.
- [10] Ghasemi, Mir Majid, Amir Fathi, Morteza Mousazadeh, and Abdollah Khoei, "A new high speed and low power decoder/encoder for Radix-4 Booth multiplier," International Journal of Circuit Theory and Applications, 2021.
- [11] Chang, Yen-Jen, Yu-Cheng Cheng, Shao-Chi Liao, and Chun-Huo Hsiao, "A Low Power Radix-4 Booth Multiplier with Pre-Encoded Mechanism," IEEE Access 8, 2020.
- [12] Akshitha V. Ramesh et al. "Implementation and Design of FIR Filters using Verilog HDL and FPGA" Perspectives in Communication Embedded-systems and Signal-processing- PiCES vol. 4.5 pp. 85-88, 2020.

- [13] Ranjeeta Yadav Rohit Tripathi and Sachin Yadav "FPGA Implementation of Efficient FIR Filter" International Journal of Engineering and Advanced Technology (IJEAT) vol. 9, no. 3 February 2020.
- [14] Zhang, Tingting, Weiqiang Liu, Jie Han, and Fabrizio Lombardi, "Design and Analysis of Majority Logic Based Approximate Radix-4 Booth Encoders," In IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH), pp. 1-6. IEEE, 2019.
- [15] Gharabaghlo, Nader Sharifi, and Tohid Moradi Khaneshan, "Performance Analysis of High Speed Radix-4 Booth Encoders in CMOS Technology," Majlesi Journal of Electrical Engineering 13, no. 3, 49-57, 2019.
- [16] Madugula Sumalatha Panchala Venkata Naganjaneyulu and K. Satya Prasad "Low power and low area VLSI implementation of vedic design FIR filter for ECG signal de-noising" Microprocessors and Microsystems vol. 71 pp. 102883, 2019.
- [17] Oguzhan COSKUN and A. V. C. I. Kemal "FPGA Schematic Implementations and Comparison of FIR Digital Filter Structures" Balkan Journal of Electrical and Computer Engineering vol. 6.1 pp. 20-28, 2018.
- [18] D. Kalaiyarasi and M. Saraswathi, "Design of an Efficient High Speed Radix-4 Booth Multiplier for both Signed and Unsigned Numbers", 4th International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB), IEEE 2018.