**How to Cite:**
Sathya Narayanan, V., Siddanthan, K., Sneha Priya, K. C., & Valarmathi, B. (2022). VLSI implementation of booth multiplier and carry select adder based fir filter design for ECG signal denoising. *International Journal of Health Sciences*, *6*(S4), 10631–10645. https://doi.org/10.53730/ijhs.v6nS4.11397

# VLSI implementation of booth multiplier and carry select adder based fir filter design for ECG signal denoising

**Sathya Narayanan V**
Department of Electronics and Communication Engineering, Kongu Engineering College, Perundurai, Tamil Nadu, India
Corresponding author email: sathya198823@gmail.com

**Siddanthan K**
Department of Electronics and Communication Engineering, Kongu Engineering College, Perundurai, Tamil Nadu, India

**Sneha Priya K C**
Department of Electronics and Communication Engineering, Kongu Engineering College, Perundurai, Tamil Nadu, India

**Valarmathi B**
Department of Electronics and Communication Engineering, Kongu Engineering College, Perundurai, Tamil Nadu, India

***Abstract*---**Over the last two decades, FIR filters have been the subject of intense research. The design of an adder, which is a major building component in circuit design, determines the overall performance of a system. The Finite Impulse Response (FIR) filter has been increasingly popular in signal processing applications in recent years. For signal processing field applications and VLSI systems, many adders are implemented. Signal denoising, as well as the production of an effective multiplier, had never been explained in any of the previous publications. This paper proposes 8 bit booth multipliers for partial products. The design employs the Carry select and Booth techniques. For partial product addition, the Carry Select Adder (CSA) is employed. The architecture of the FIR filter is proposed for operation with Electro Cardiogram (ECG) signal. It's known as CSA-BOOTH FIR, and it's used for denoising. Using the MATLAB application, the ECG signal with noise is sent into the filter. The denoising method is written in Verilog, with the output recorded in a text file. The binary values are read in MATLAB to denoise the signal. The performance of FPGAs and ASICs is evaluated. Signal to Noise Ratio (SNR), Bit Error

Rate (BER), and Mean Square Error (MSE) are all calculated from denoised data.

***Keywords*---**VLSI, FIR filter, carry select adder, ECG, FPGA, SNR, LUT, MSE, adder, multiplier.

## Introduction

In signal processing applications, FIR filters are now commonly employed. The FIR filter is used in FPGA to analyse the architecture's hardware use. In the building of a FIR filter, the multiplier is very significant. The whole architecture uses more area to execute the filter operation when the FIR filter incorporates additional multipliers. As a result, in FIR filter design, minimising the area in multipliers is a critical issue. The FIR filter has still has several advantages, including multi-rate processing efficiency, linear phase response, and a desirable numerical feature for finite precision and fractional arithmetic. A digital multi-standard FIR filter is the FIR filter. System generator, normal FIR filter, Sensitivity Driven Algorithm(SDA), linear phase FIR filter, Least Mean Square(LSM) algorithm based FIR filter, and DA based FIR filter are some of the existing architectures for FIR filter design.

Hardware consumption is inefficient in all architectures. Furthermore, applications are not prioritised in traditional systems. To overcome this problem, this paper offers the Carry Select adder-based FIR filter (CSAFIR) filter technique. The processing element requires the input and co-efficient elements to be completed. The employment of Booth multiplier and CSA improves the performance of FPGAs and ASICs. The usage of carry choice adders reduces propagation time since the adder block does not need to wait for the carry to complete the addition process. Instead, the carry select adder computes the output together with carry based on the input carry and executes the operation to produce carry in a separate block.

FIR filters have a finite lifetime since their impulse response, or reaction to any finite length input or succession, settles to zero in a finite amount of time. Before settling to zero, the impulse response of the Nth order discrete time FIR filter exceeds N+1 samples. FIR filters can be discrete or continuous time, digital or analogue, and digital or analogue in nature. The advantage of FIR filters is that they do not require feedback. This means that cumulative iterations do not compound rounding errors. Each calculation has the same level of relative error. This makes the installation process easier. They are intrinsically stable since the outcome is the sum of a finite f - number. System generator, normal FIR filter, Decimal FIR filter, linear phase FIR filter, GA based FIR filter, parallel based FIR filter, low power multiplier FIR filter, and DA based FIR filter are samples of FIR filter architectures. All of the architectures have inefficient hardware consumption. Furthermore, these existing systems do not prioritise applications.

To overcome this problem, this paper offers the booth carry look ahead adder-based FIR filter (BM-CLAFIR). The elements are needed to conduct the processing element input and co-efficient. With the help of the booth multiplier, those two

inputs are multiplier inputs. The addition procedure is performed by the carry look ahead adder (CLA) in the accumulator module. The proposed method outperforms traditional methods in terms of FPGA and ASIC performance thanks to the use of BM and CLA. Carry select adders are utilised in booth multiplier architecture.

The objective to use carry select adders is that the propagation delay is reduced as the adder block need not wait for the carry, to perform the addition process instead the carry select adder performs the operation to generate carry in a separate block and based on the input carry given the adder computes the output along with carry. The human heart is an organ that pumps blood through the circulatory system's veins, giving oxygen and nutrition to the tissues while also eliminating carbon dioxide and other pollutants.

An electrocardiogram (ECG) is a fast test that used to evaluate your heart's rhythm and electrical activity. Sensors are attached to your skin detect the electrical signals produced by your heart each time it beats. An ECG records the electrical signals in your heart. It's a painless and quick test for diagnosing cardiac issues and monitoring your heart's health. Modern ECG sensors record ECG signals in a bandpass range of 0.05 (or 0.5) Hz to 100 (or 150) Hz.

**Existing method**

**Vedic design CLA based fir filter**

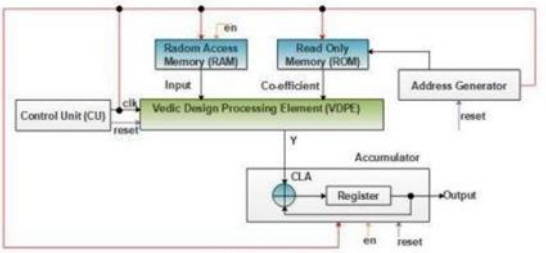



Figure 1 VD-CLA-FIR filter

The VD-CLA-FIR filter architecture's address generator, accumulator, clock generator, register, Read Only Memory (ROM), Random Access Memory (RAM), and control unit are shown in Figure 1. The clock signal is generated by the clock generator. The coefficient data is stored in the ROM, while the noisy data is stored in the RAM. The clock signal is passed from the control unit to the filter, which calculates the filter block's registers are reset via the reset signal. The data addresses are generated by the address generator. It aids in the reading of data from ROM in order to complete the filter operation computation. To accomplish the filter operation, the input and co-efficient are read from RAM and ROM, respectively. To construct the FIR filter outputs (y=0), the noisy data value is used as an input (0, 255, etc.) and the coefficient (213, 18, etc.) is multiplied by the coefficient. The FDA tool in MATLAB was used to calculate the coefficient. PE was developed following Vedic design principles, which are said to reduce hardware usage. The accumulator's register is initially set to zero. The PE output is added

to this register value, which is likewise maintained in the same register. Finally, the register's The output has now been delivered.

**Block diagram of the processing element**

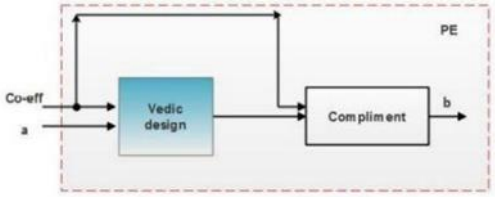



Figure 2 Block Diagram of PE

The accumulator's register is initially set to zero. The PE output is added to this register value, which is likewise maintained in the same register. Finally, the register's output is delivered. Eq defines the common form of the N-tap FIR filter (1).

$$y(n) = \sum_{k=0}^{N-1} h(k)x(n-k)$$

In this case, n equals 0, 1, 2,..., and so on. The output of the FIR filter is y(n), and the coefficient of the FIR filter is y. (n). The number of input sequences is represented by x. (n k). Figure 2 depicts the Processing Element (PE), one of the most important components in FIR filter building. The remaining block, according to the PE, is performing brilliantly. The Vedic design multiplies and processes an input and coefficients. Figure 2 depicts the Processing Element (PE), which is one of the most crucial pieces of a FIR filter.

**Vedic design algorithm**

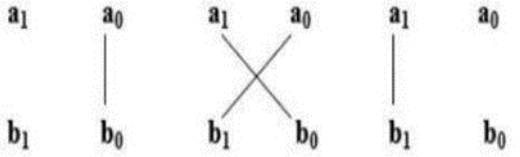



Figure 3 The 2x2 Multiplication by using a Vedic Multiplier

Figure 3 explains how to multiply two numbers by two using a Vedic multiplier. When compared to ordinary multipliers, The Vedic Multiplier architecture executes at a breakneck speed. In all forms of number schemes, the Vedic multiplier is used. To show how Urdhva Tiryakbhyam can be multiplied. Take the multiplicand (a1, a0) and multiplier (a1, a0) as examples (b1, b0). As a result, the binary number multiplication method produces a four-bit output. In general, the Vedic multiplier follows the steps outlined below.

Step 1: Multiply the multiplicand's Least Significant Bit (LSB) with the multiplier vertically to get the LSB's ultimate value.
Step 2: Add the products by multiplying the multiplicand's LSB with the multiplier's Most Significant Bit (MSB) and the multiplicand's MSB with the multiplier's LSB (crosswise). The second element of the ultimate result is determined by the adding process.
Step 3: Multiply the multiplicand's MSB by the multiplier (vertically). The item is added to the previously obtained carry. The third and fourth bits of the final result are used to calculate the total and carry.

**Carry look-ahead adder design**

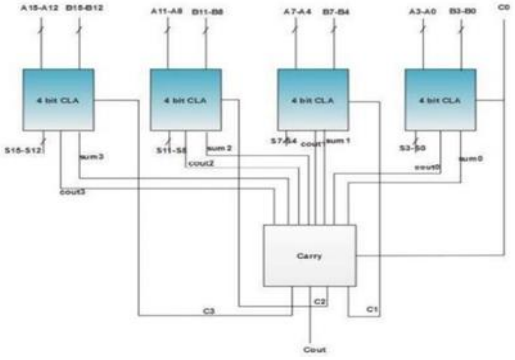



Figure 4 Block Diagram of the 16-bit

**CLA Design**

The VD-CLA-FIR filter architecture in this study uses a 16-bit CLA instead of a traditional adder, as seen in Figure 4. To improve system performance, this CLA is utilized in the accumulator module. By performing speedy arithmetic operations in a variety of data processing methods, this adder saved space and power.

**BIT CLA ADDER**

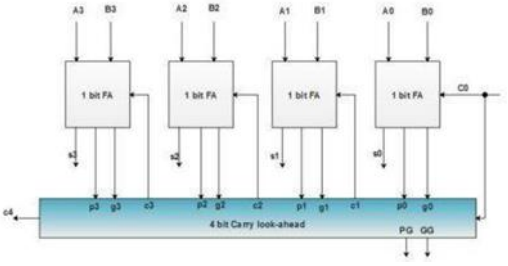



Figure 5 4 Bit CLA Adder

Figure 5 depicts the 16-bit CLA, which is made up of four 4-bit CLA blocks and a carry generator. A 4-bit CLA is forced to drive all of the P and G internal signals, culminating in a 16-bit CLA. CLA adders are usually constructed as 4-bit modules, which are commonly utilised to construct large adders. The power,

latency, and area of the VD-CLA-FIR approach can all be lowered by using 16-bit CLA.

**Proposed method**

**Overall block diagram**

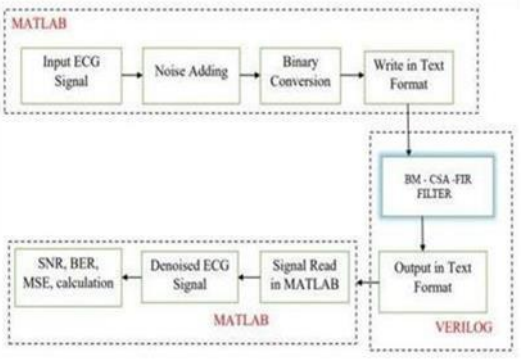



Figure 6 Overall Block Diagram

Step 1: The ECG signal is read from an online source using MATLAB and the arrhythmia database.
Step 2: The ECG signal is masked with white Gaussian noise (WGN). Between 0.1 and 0.5 will be the noise density. To add noise to an input signal in MATLAB, use the "awgn" function.
Step 3: The dec2bin MATLAB tool is used to convert the WGN with ECG signals to binary format.
Step 4: In step four, save the binary signal to a text file (example: Noisy ECG signal.txt).
Step 5: The Noisy ECG signal.txt file is fed into Verilog's input. In the majority of cases, the input value was generated at random. This noisy binary value, on the other hand, is now considered an input and will be saved in RAM. In the read-only memory, the coefficient is preserved (ROM).
Step 6: The Booth Multiplier-Carry Select Adder-FIR (DM-CSA-FIR) filter is used to minimize noise in Verilog.
Step 7: The results of each clock cycle accumulator are saved in a text file. (For example, filter output.txt.) The Verilog code can be used to assess the ASIC and FPGA performance (area, power, and latency) (LUT, flip flops, slices, and frequency).
Step 8: To obtain denoising ECG data, open the Filter output.txt file in MATLAB.
Step 9: MSE, SNR, and BER are determined from the denoising ECG signal. 8T SRAM cell based on CMOS inverter is replaced by PNN inverter.

**Carry select adder and booth multiplier**

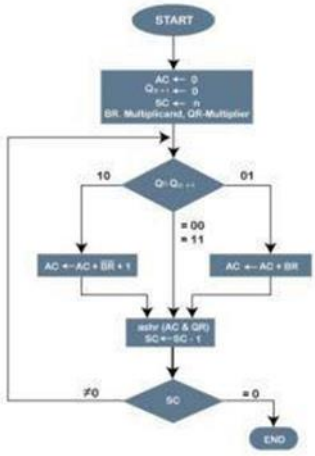



Figure 7 Booth Multiplier

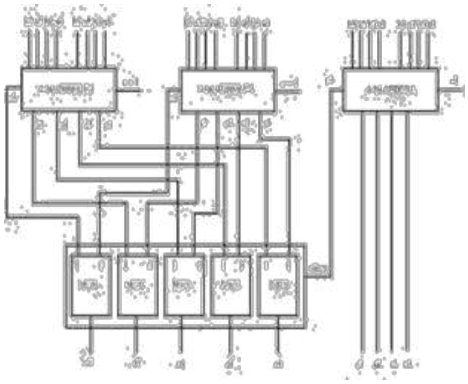


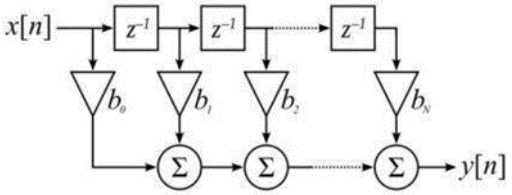
Figure 8 Carry Select Adder

**FIR Filter**

There are two kinds of filters, they are FIR (Finite Impulse Response) and IIR (Inverse Impulse Response) (Infinite Impulse Response). A signal processing filter with a finite impulse response (FIR) settles to zero in a finite amount of time is known as a finite impulse response (FIR) filter (or reaction to any finite length input). With infinite impulse response (IIR) filters, which can respond indefinitely, internal feedback is feasible (usually decaying). A Nth-order discrete-time FIR filter's impulse response (that is, the output in response to a Kronecker delta input) lasts precisely N+1 samples (from first nonzero element to last nonzero element) before settles to zero. There are discrete-time and continuous-time FIR filters, as well as digital and analogue versions. Finite impulse response filters are categorized as low pass FIR filters, high pass FIR filters, band pass FIR filters, and band stop FIR filters based on the frequencies in which the input sample must be filtered.

**Structure of FIR filter**

A digitally constructed finite impulse response (FIR) filter can implement practically any frequency response. A FIR filter is often created using a series of delays, multipliers, and adders to generate the filter's output. Previous input samples are utilized as a result of the delays. The multiplication coefficients are denoted by the hk values, whereas the output at time n is the sum of all delayed samples multiplied by the relevant coefficients. The process of calculating the length and coefficients of a filter is known as filter design. The goal is to set those values so that when the filter is executed, it produces certain desirable stopband and passband characteristics. The type of filter and the order of the filter, which may be in the form of nth order, are used to build a FIR filter. The coefficient of the filter to be designed must be known before it can be designed. Components in filters include multipliers, adders, and delay elements (flip flops).




**Results and Discussion**

**MATLAB Result**

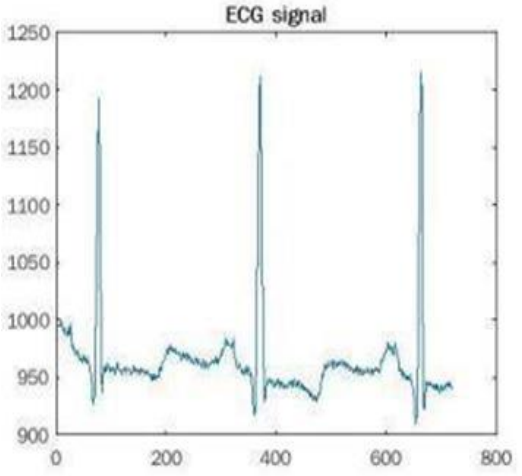



Figure 9 ECG Signal

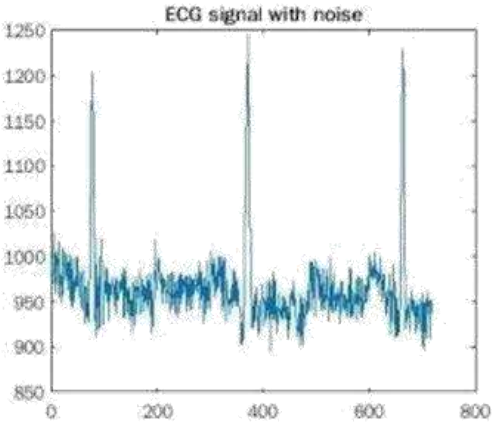



Figure 10 ECG Signal with Noise

The input signal ECG is taken from the internet resource and plotted using MATLAB to see the waveform. After the reading the ECG signal from the dataset the noise signal has to be added and the noise added ECG signal as shown in figure 9 and figure 10.

**Simulation Results**

**Carry Select Adder**

The simulation is performed in Verilog to verify the functionality of conventional full adder and synthesized using Model sim. The simulation result of 16X16 carry select adder is shown in the figure 11. Two 16 bit data is given as an input and the 16 bit output is obtained.

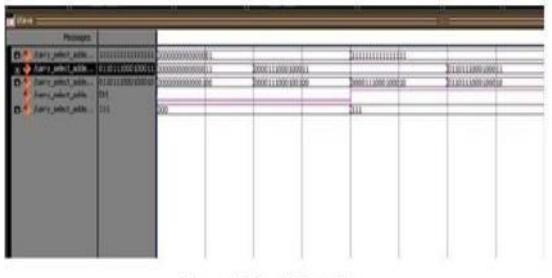


Figure 11 Carry Select Adder

**8X8 Booth Multiplier**

The simulation is performed in Verilog to verify the functionality of conventional full adder and synthesized using Model sim. Figure 12 depicts the simulation result of an 8X8 Booth multiplier. Two 8bit data is given as an input. The input values are '7' and '3' in an 8bit binary format as X and Y and the output is obtained as an '21' as P.


Figure 12 Booth Multiplier

**FIR Filter**

The simulation of FIR is performed in Verilog to verify the functionality and effective denoising process of low pass 12-TAP Filter and synthesized using Model sim. Figure 13 depicts the FIR Filter simulation result.

Figure 13 FIR Filter

**Implementation Results**

**FIR filter device utilization**

The FIR Filter with Booth multiplier and carry select adder constructed utilizing the Spartan3E FPGA device is shown in Figure 14 with its area, resource use, and delay report.

| Device Utilization Summary | | | | |
|---|---|---|---|---|
| Logic Utilization | Used | Available | Utilization | Note(s) |
| Number of Slice Flip Flops | 151 | 66,560 | 1% | |
| Number of 4 input LUTs | 3,563 | 66,560 | 5% | |
| Number of occupied Slices | 1,919 | 33,280 | 5% | |
| Number of Slices containing only related logic | 1,919 | 1,919 | 100% | |
| Number of Slices containing unrelated logic | 0 | 1,919 | 0% | |
| Total Number of 4 input LUTs | 3,563 | 66,560 | 5% | |
| Number of bonded IOBs | 114 | 784 | 14% | |
| Number of MULT18X18s | 22 | 104 | 21% | |
| Number of BUFGMUXs | 1 | 8 | 12% | |
| Average Fanout of Non-Clock Nets | 3.91 | | | |

Figure 14 Device Utilization of FIR Filter

Device utilization summary:
---
Selected Device : 3s5000fg1156-4
Number of Slices: 1842 out of 33280 5%
Number of Slice Flip Flops: 151 out of 66560 0%
Number of 4 input LUTs: 3559 out of 66560 5%
Number of IOs: 123
Number of bonded IOBs: 114 out of 784 14%
Number of MULT18X18s: 22 out of 104 21%
Number of GCLKs: 1 out of 8 12%

The delay report of

Delay: 32.177ns (Levels of Logic = 22)
Source: delay_SRL_7_3 (FF)
Destination: FIR_TAP_OUT_15 (FF)

**RTL design**

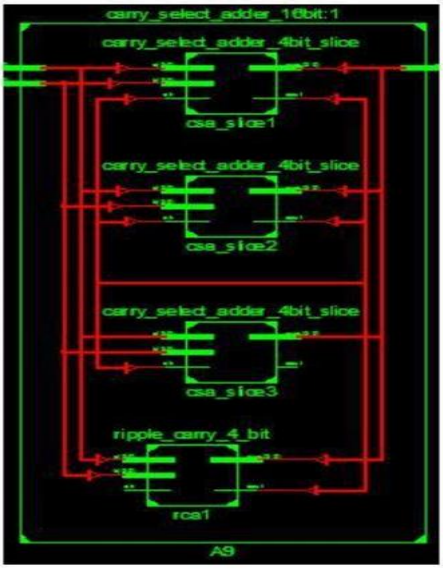



Figure 15 RTL of Carry Select Adder

**Schematic Diagram**

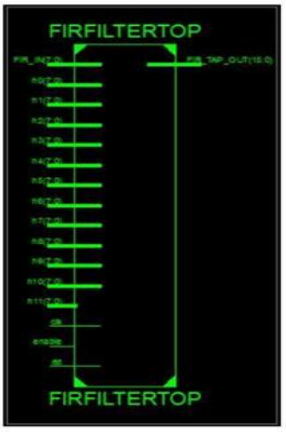



Figure 16 RTL of FIR Filter

Figure 15, Figure 16 and Figure 16 Shows the RTL design which was designed with the help of Xilinx

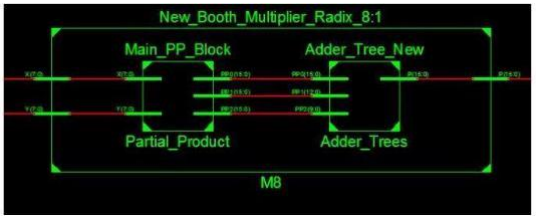



Figure 17 RTL Design of Booth Multiplier

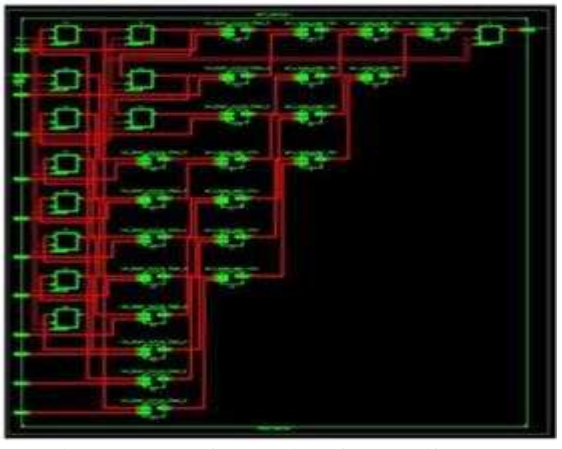


Figure 18 Schematic of FIR Filter

Figure 18 shows the schematic diagram of the FIR Filter produced from the Xilinx tool after simulation.

## Conclusion

The VD-CLA-FIR filter was developed in Xilinx using Verilog code and is focused on the Booth Multiplier with the Carry Select Adder method. The ECG signals are first analysed in MATLAB, which is susceptible to noise. Then the signal is converted into binary text file using MATLAB. The multiplication operation was carried out via the Booth Multiplier method in this project. The FIR filter is applied to the ECG signal to minimise noise. Because the delay computed by the Booth multiplier is smaller than the delay computed by the conventional design, the overall performance of the FIR filter has been improved by employing the Booth multiplier architecture instead of the regular multiplication architecture. The

Carry Select adder solves the problem of propagation delay, resulting in a more effective filter design. As a result, the FIR filter is created in ModelSim, and the filter response is transformed to the frequency domain using FFT for the time domain output. From the response, it is concluded that the frequencies greater than 60Hz which are noise signals have been attenuated. The FIR filter implementation results show that 5 percent of the LUT is used, and 1 percent of the flip flop is used. The FIR filter's overall power consumption is measured to be 75.723 W.

## References


1. Andrea Bonetti, Adam Teman, Philippe Flatresse, Andreas Burg, (2017) 'Multipliers driven perturbation of coefficients for low-power operation in reconfigurable FIR filters', IEEE Trans. Circuits Syst. I Vol.64, No.9,PP. 2388–2400.
2. B. Doss, K. Soundararajanm, Y. Narasimha Murthy, (2015) 'Low-Power and low-area adaptive FIR filter based on DA using FPGA', Int. J. Sci. Res. Manage. Vol.3, No.1.
3. B. Rashidi, B. Rashidi, M. Pourormazd, (2011) 'Design and implementation of low power digital FIR filter based on low power multipliers and adders on xilinx FPGA', in: Electronics Computer Technology (ICECT), 2011 3rd International Conference on, 2, IEEE, , April, PP. 18–22.
4. C. Xu, S. Yin, Y. Qin, H. Zou, (2013) 'A novel hardware efficient FIR filter for wireless sensor networks. in ubiquitous and future networks (ICUFN)',Fifth International Conference on, IEEE, July, PP. 197–201.
5. Chia-Yu Yao, Wei-Chun Hsia, Yung- Hsiang Ho, (2014) 'Designing hardware efficient fixed-point FIR filters in an expanding subexpression space', IEEE Trans. Circuits Syst. I Vol.61,No.1, PP. 202– 212.
6. Dhanumjaya.K,Sudha.M, Dr.MN.Giri Prasad, Dr.K.Padmaraju (2012) 'Cell Stability Analysis of Conventional 6T Dynamic 8T Sram Cell in 45nm Technology', International Journal of VLSI design & Communication Systems (VLSICS), Vol.3, No.2, pp.41.
7. Iqbal, J.L. Mazher, S. Varadarajan, (2013)'Highperformance reconfigurable FIR filter architecture using optimized multiplier', Circuits Syst. Signal Process. Vol.32, No.2, PP.663– 682.

8. Jiajia Chen, Jinghong Tan, Chip-Hong Chang, Feng Feng, (2016) 'A new cost-aware sensitivity driven algorithm for the design of FIR filters', IEEE Trans. Circuits Syst. I PP 99 1–11.
9. Madugula Sumalatha, Panchala Venkata Naganjaneyulu, Kodati Satya Prasad, (2018) 'Low power and low area VLSI implementation of Radix- 8 Carry look ahead adder FIR filter for DSP applications', J. Adv. Res. Dyn. Control Syst. Vol.10, PP.70– 82.
10. Martin Kumm, Konrad Möller, PeterZipf,(2013) 'Reconfigurable FIR filter using distribute arithmetic on FPGAs', Circuits and systems (ISCAS), 2013 IEEE International Symposium on IEEE,.
11. S. Bhattacharjee, S. Sil, A. Chakrabarti, (2013) 'Evaluation of power efficient FIR filter for FPGA based DSP applications', ProcediaTechnol.Vol.10, PP.856–865..S.J. Lee, J.W. Choi, S.W. Kim, J. Park, (2011) 'A reconfigurable FIR filter architecture to trade off filter performance for dynamic power consumption', IEEE Trans. Very Large Scale Integr. VLSI Syst. Vol.19, No.12, PP. 2221–2228..
12. S. Harize, M. Benouaret, N. Doghmane, (2013) 'A methodology for implementing decimator FIR filters on FPGA', AEU-Int. J. Electron. Commun. Vol.67, No.12, PP.993– 1004.
13. S. Khan, Z.A. Jaffery, (2015) 'Low power FIR filter implementation on FPGA using parallel distributed arithmetic', in: India Conference (INDICON), 2015 Annual IEEE, IEEE, , December, pp. 1–5..
14. S. Ramanathan, Gorty Anand, Prasanth Reddy, Sri Adibhatla Sridevi, (2016) 'Low power adaptive FIR filter based on distributed arithmetic', Int. J. Eng. Res. Appl. Vol.6, PP. 47– 51.
15. Sang Yoon Park, and Pramod Kumar Meher,(2015) 'Efficient FPGA and ASIC realizations of DA Based reconfigurable FIR digital filter', IEEE Transactions on Circuits and Systems II: Express Briefs.
16. Sang Yoon Park, Pramod Kumar Meher, (2013) 'Low-power, high-throughput, and low area adaptive FIR filter based on distributed arithmetic', IEEE Trans. Circuits Syst. II Vol.60, No.6, PP. 346–35.
17. Sang Yoon Park, Pramod Kumar Meher, (2013) 'Low-power, high-throughput, and low area adaptive FIR filter based on distributed arithmetic', IEEE Trans. Circuits Syst. II Vol.60, No.6, PP. 346–35.
18. Susilo, C. B., Jayanto, I., & Kusumawaty, I. (2021). Understanding digital technology trends in healthcare and preventive strategy. *International Journal of Health & Medical Sciences*, *4*(3), 347-354. https://doi.org/10.31295/ijhms.v4n3.1769
19. U. Meyer-Baese, G. Botella, D.E. Romero, M. Kumm, (2012) 'Optimization of high speed pipelining in FPGA-based FIR filter design using genetic algorithm', International Society for Optics and Photonics, May, p. 84010R..
20. V. Sathyanarayanan, (2020) 'Noninvasive Technique for Diabetic Prediction Using ECG and PPG Signal Parameters', International Journal of Advanced Science and Technology. Vol.29, PP. 6504 –6512.
21. Widana, I.K., Sumetri, N.W., Sutapa, I.K., Suryasa, W. (2021). Anthropometric measures for better cardiovascular and musculoskeletal health. *Computer Applications in Engineering Education*, *29*(3), 550–561. https://doi.org/10.1002/cae.22202

22. X. Lou, P.K. Meher, Y. Yu, W. Ye, (2017) 'Novel structure for area-efficient implementation of FIR filters', IEEE Trans. Circuits Syst. Express Briefs Vol.64, No. (10), PP.1212–1216.
23. Y.C. Tsao, K. Choi, (2012) 'Area- efficient VLSI implementation for parallel linearphase FIR digital filters of odd length based on fast FIR algorithm', IEEE Trans. Circuits Syst. II Vol.59, No.6, PP. 371– 375.