Evolution in Electrical and Electronic Engineering Vol. 4 No. 1 (2023) 347-355
© Universiti Tun Hussein Onn Malaysia Publisher's Office

EEEE

Homepage: http://publisher.uthm.edu.my/periodicals/index.php/eeee
e-ISSN : 2756-8458

Hardware Implementation of an FIR Filter For Electrocardiogram (ECG) Signal Processing

Mohd Izzul Ariffi Ardani[1], **Chessda Uttraphan**[1]*

[1]Departmant of Electronic Engineering, Faculty of Electrical and Electronic Engineering,
Universiti Tun Hussein Onn Malaysia, Batu Pahat, 86400, MALAYSIA

*Corresponding Author Designation

DOI: https://doi.org/10.30880/eeee.2023.04.01.041
Received 15 January 2023; Accepted 10 April 2023; Available online 30 April 2023

Abstract: Electrocardiogram (ECG) signals are the heart activity electrical signals that are used to diagnose heart related diseases. Though, the noise that is induced to the original ECG signal during the measurement procedure might affect the accuracy of the diagnosis. Today, digital filters such as finite impulse response (FIR) filters are used to remove noise from signals. However, the speed of the FIR filter implemented in the software is slow if the ECG data is large, since the processing is performed serially in software implementation. This work proposes the hardware design of an ECG FIR filter to speed up the denoising process. The ECG data is obtained from the Physionet database and MATLAB is used to determine the filter coefficients using the Kaiser Window method. The design is then mapped into the hardware logics and is modelled in Verilog Hardware Description Language (HDL), where the targeted hardware is the Intel Cyclone IV Field Programmable Gate Array (FPGA). Simulation results from MATLAB and FPGA implementations were obtained and analyzed. Analysis of power consumption, logics utilization, and maximum operating frequency were also conducted. The results show that the proposed design implemented on an FPGA can process the ECG signal up to 260 times faster than MATLAB due to parallel processing that can only be realized in hardware implementation. The logic utilization and power consumption can be further improved by implementing operation scheduling and constrained resource allocation.

Keywords: Electrocardiogram, Finite Impulse Response, Kaiser Window, Intel Cyclone IV, Field Programmable Gate Array

1. Introduction

ECGs are the most useful cardiac monitoring tool because they show the heart's condition as a bio-electric potential waveform [1]. Even a small amount of noise can change the ECG signal [2]. FIR filters are widely used due to the high noise immunity, accuracy, easy filter characteristics modification, and component variation freedom. The FIR filter is currently be implemented on a software environment, running on a general-purpose processor. The digital signal processing needs a lot of computation;

**Corresponding author: chessda@uthm.edu.my* therefore, it will take relatively longer time to produce the outputs if it is implemented using the general-purpose processor because of limited hardware resources.

An FPGA offers customable logic configurations, hence, able to create parallel data processing hardware. With increasing size, FPGAs have more programmable logic cells and routing resources [2] –[4]. Kaiser Window will be utilized in this work for designing the FIR filter.

1.1 FIR Filter Classification

Designing the FIR filter is a typical method for filtering a noisy signal. Since the ECG signal has a frequency range between 0.5 Hz and 50 Hz [5] and the frequency that greater than 50 Hz is considered as noise. Furthermore, if the signal frequency is the same as the noise frequency, the use of the digital filter to reduce noise will be affected and highly ineffective [6][7]. Recently, software and hardware implementation on reducing the noise have been purpose [8]. For example, Sumbal Zahoor and Shahzad Naseem purpose in design a digitalized FIR filter using the Kaiser Window is the best choice because the window can demonstrate both a narrow transition band and a wider main lobe width [9].

Besides, the implementation in using low pass filter have been purpose where an FIR low pass filter can reduce noise involves the repetitive application of a particular filter logic and the design will require more space and more power to process it [10].

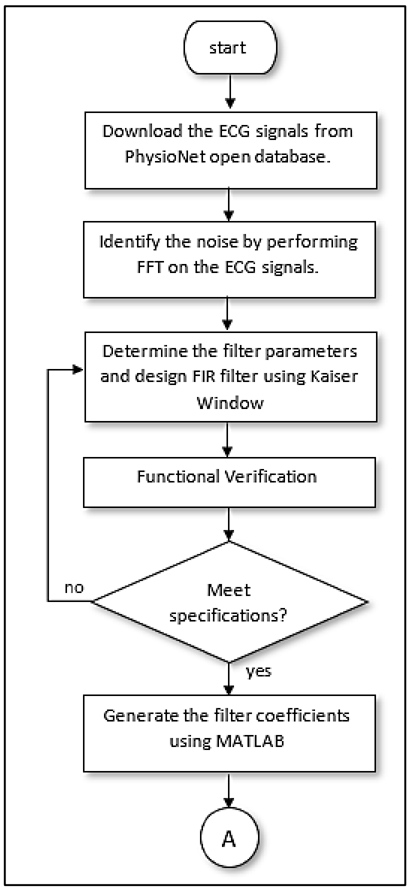
2. Materials and Methods

The work implements the software and hardware algorithms, where the MATLAB is used to design the FIR filter using the Kaiser Window method. The coefficients that were obtained from MATLAB were used to produce the signal data flow graph (SDFG), where it is then mapped into a hardware logic design.

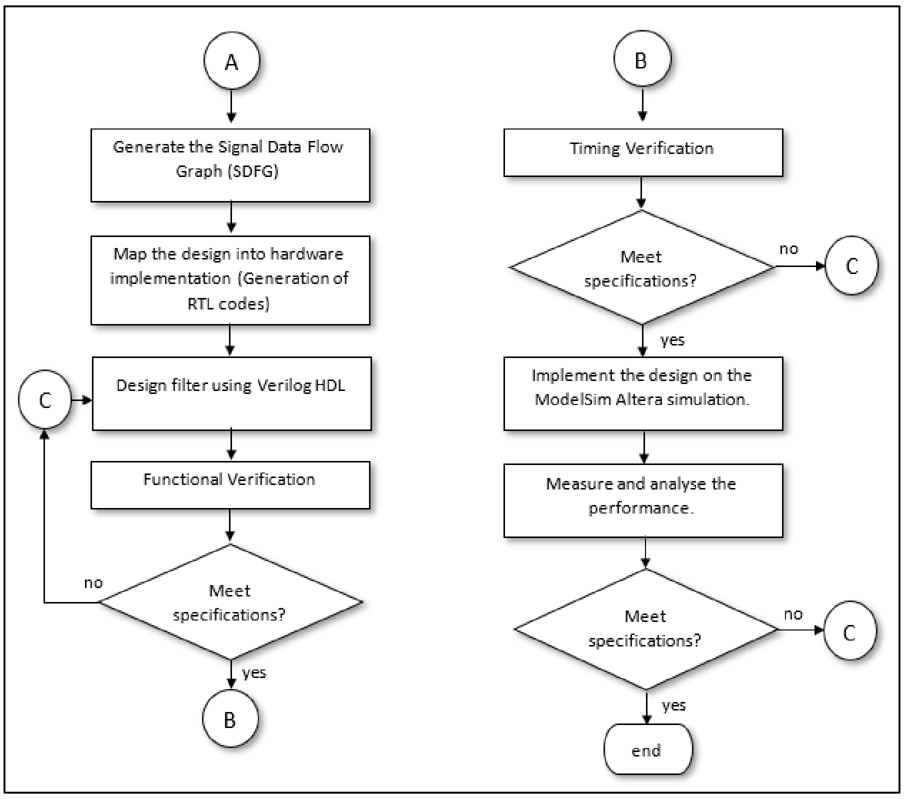
2.1 Flow Charts

Figure 1(a) shows a flow chart of the FIR filter for ECG signal filtering design process which comprises of a noisy ECG signal, an FFT one-sided spectrum, and an FIR filtering design that uses the Kaiser Window method. The noisy ECG signal will be imported into MATLAB, where the FFT will be performed in order to determine the noise power components. After identifying noise components in the FFT frequency spectrum, the Kaiser window is used to design the FIR low pass filter that will remove high frequency components from the ECG signal.

As shown in Figure 1(b), the filter coefficients that were obtained from MATLAB will be used in logic design for FPGA implementation. The coefficients are converted to a 16-bit unsigned fixed-point values. By taking the coefficient value, the filter will generate a Signal Data Flow Graph (SDFG) in floating point and mapping the design into hardware implementation where in fixed point. The ModelSim Altera simulator displays the filtered ECG signal as a fixed-point unsigned integer and the simulation will measure and analyse the time performance.


start
Download the ECG signals from PhysioNet open database.
Identify the noise by performing FFT on the ECG signals.
Determine the filter parameters and design FIR filter using Kaiser Window
Functional Verification
Meet specifications?
no
yes
Generate the filter coefficients using MATLAB
A

(a)


A
Generate the Signal Data Flow Graph (SDFG)
Map the design into hardware implementation (Generation of RTL codes)
C
Design filter using Verilog HDL
Functional Verification
Meet specifications?
no
yes
B
B
Timing Verification
Meet specifications?
no
C
yes
Implement the design on the ModelSim Altera simulation.
Measure and analyse the performance.
Meet specifications?
no
C
yes
end

(b)

Figure 1: Flow Chart (a) Filtering model design (b) FPGA hardware implementation

2.2 MATLAB and FPGA Findings

Figure 2(a) depicts three sequences for designing the filtering model in MATLAB: pre-processing, processing, and post-processing. The process begins with loading a data sample from PhysioBank and identifying the power noise frequency using the Fast Fourier Transform (FFT). In the subsequent phase of design, the FIR filter coefficient value in the Z transform transfer function equation is obtained using the Kaiser Window method, as shown in Eq.1. At last, the phases that produce an ECG signal output with the smoothest response.

$$H(z) = \frac{Y(z)}{X(z)} = b_0 + b_1 z^{-1} + \cdots + b_K z^{-K} \qquad Eq. 1$$

Quartus Prime will be used for hardware implementation, and there will be three modelling phases, as shown in Figure 2(b). The ROM will load the sample data as a fixed-number value. The next step involves processing the FIR filter hardware design, where all coefficient values obtained from modelling in MATLAB will be implemented to filter the ECG sample from the ROM. ModelSim Altera will conclude by simulating the hardware and analyzing the filtered ECG signal.

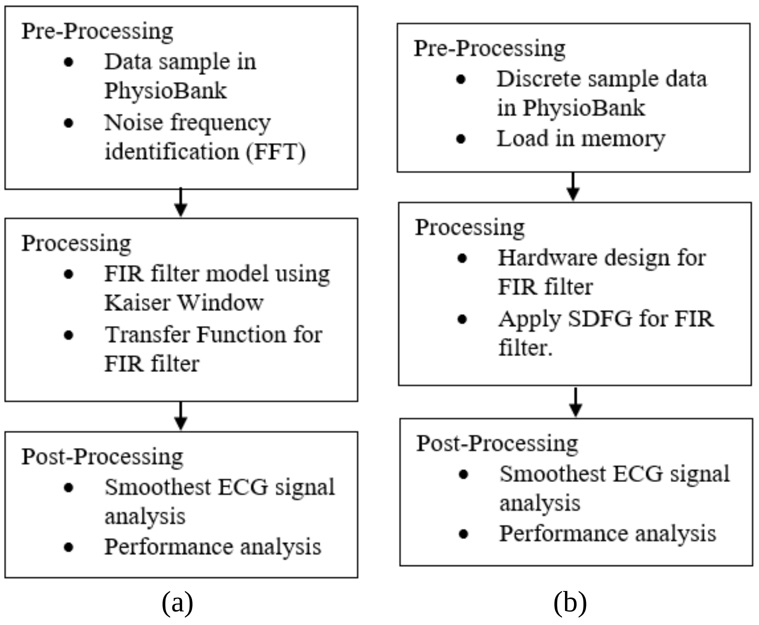

Pre-Processing
Data sample in PhysioBank
Noise frequency identification (FFT)
Processing
FIR filter model using Kaiser Window
Transfer Function for FIR filter
Post-Processing
Smoothest ECG signal analysis
Performance analysis
Pre-Processing
Discrete sample data in PhysioBank
Load in memory
Processing
Hardware design for FIR filter
Apply SDFG for FIR filter.
Post-Processing
Smoothest ECG signal analysis
Performance analysis
(a)
(b)

Figure 2: Three phase sequence modeling (a) Filtering model design using Kaiser Window (b) Hardware modelling for filtering design.

3. Results and Discussion

The implementations of the FIR filter in MATLAB and FPGA are performed and simulation outputs are presented and analysed. The performance for the FPGA implementation is analysed based on the running time in filtering the ECG signal. Logics utilization and circuit power consumption are also presented and discussed.

3.1 FIR Filter Design in MATLAB and FPGA Implementation

To design the FIR filter using Kaiser Window in MATLAB, the filter needs the specification requirement to generate the frequency domain for the signal, as shown in Table 1. The filter is required as a low-pass filter to remove the power noise in the FFT, and the minimum value for filter order in eliminate the 60 Hz frequency power noise is 20.

Table 1: MATLAB requirement for design FIR filter

Characteristic	Filter Design Specification
Response Type	Low-pass
Filter Order	20
Cut-off Frequency	20 Hz

The coefficient values will be obtained from the filter modelling in MATLAB based on the observations in Table 2, which will be useful for the hardware FIR filter design in Figure 3 and implementation in Verilog Quartus Prime. The value of the coefficient will lead to a number that is less than one. In Verilog, this operation is done by multiplying by 16 bits and then rounding the result.

Table 2: FIR filter coefficient value

	Coefficient Value, b_n	16-bit decimal value of b_n	Rounded value of b_n
0	0.004375	286.7200	287
1	0.009402	616.1695	616
2	0.016750	1097.728	1098
3	0.026340	1726.218	1726
4	0.037750	2473.984	2474
5	0.050230	3291.873	3292
6	0.062760	4113.039	4113
7	0.074170	4860.805	4861
8	0.083320	5460.460	5460
9	0.089250	5849.088	5849
10	0.091300	5983.437	5983
11	0.089250	5849.088	5849
12	0.083320	5460.460	5460
13	0.074170	4860.805	4861
14	0.062760	4113.039	4113
15	0.050230	3291.873	3292
16	0.037750	2473.984	2474
17	0.026340	1726.218	1726
18	0.016750	1097.728	1098
19	0.009402	616.1695	616
20	0.004375	286.7200	287

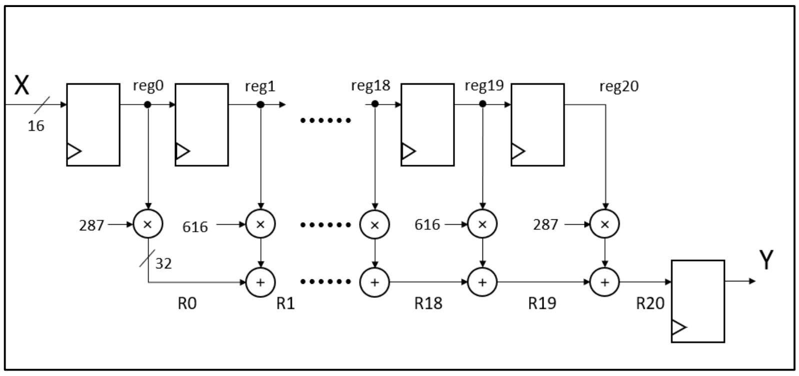


Figure 3: FBD architecture for FIR low pass filter

3.2 Result Outcome

Figure 4 depicts the ECG samples generated by MATLAB before and after filtering. In modelling the FIR low-pass filter, the outcomes are based on the Kaiser Window implementation.

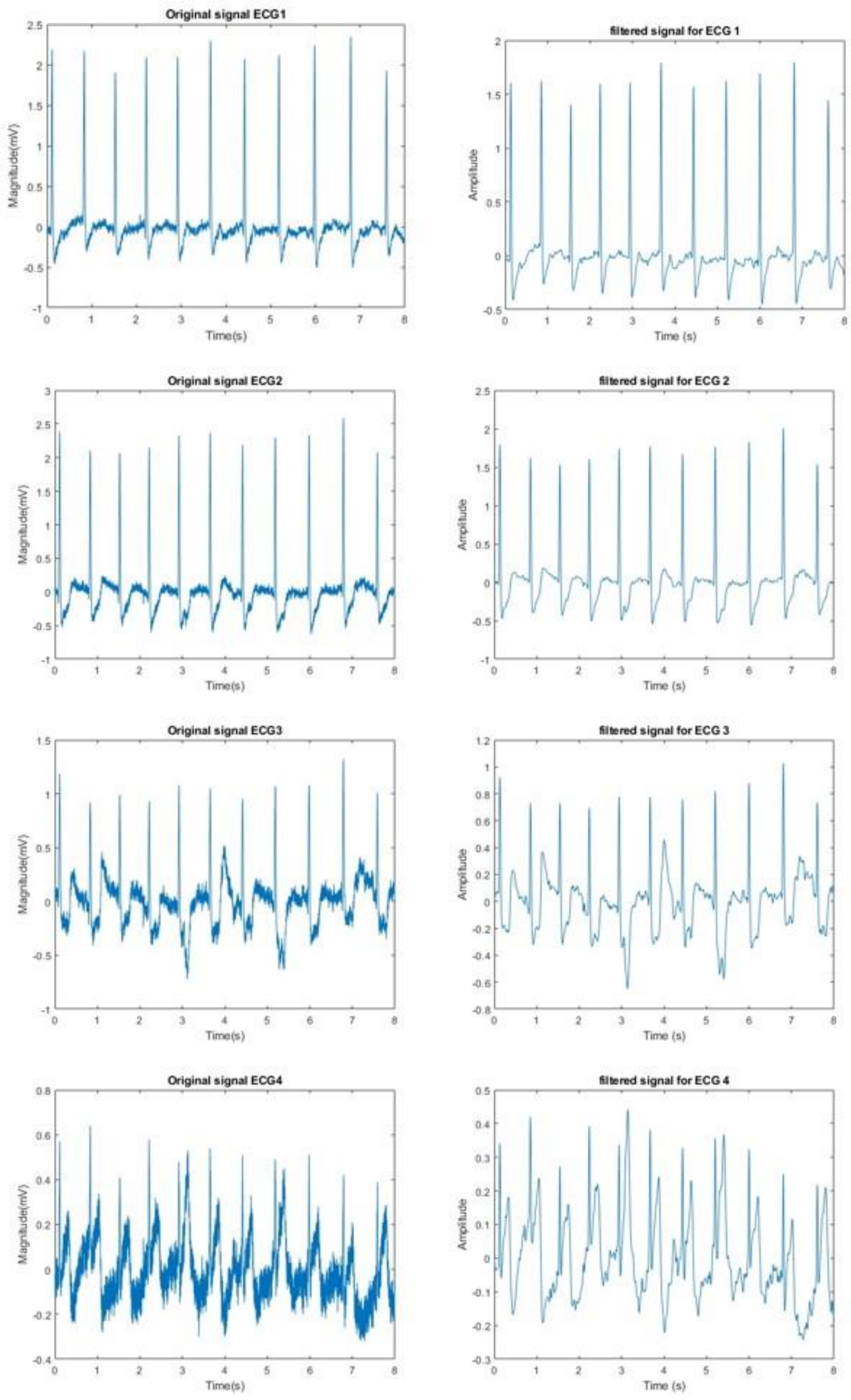
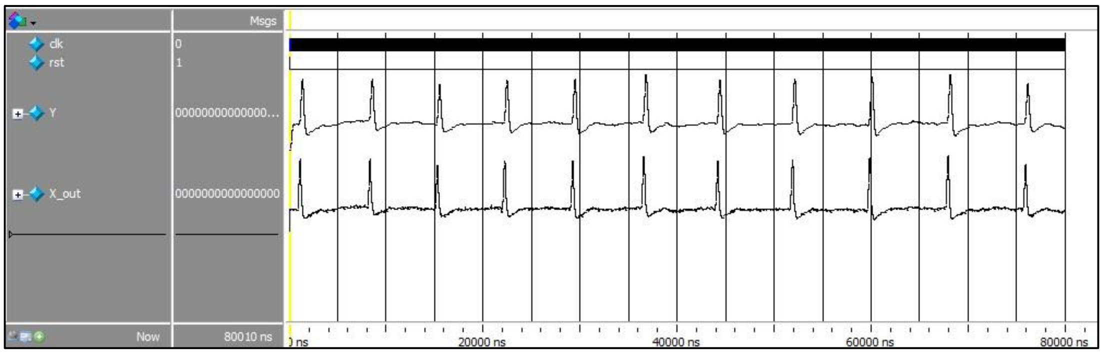
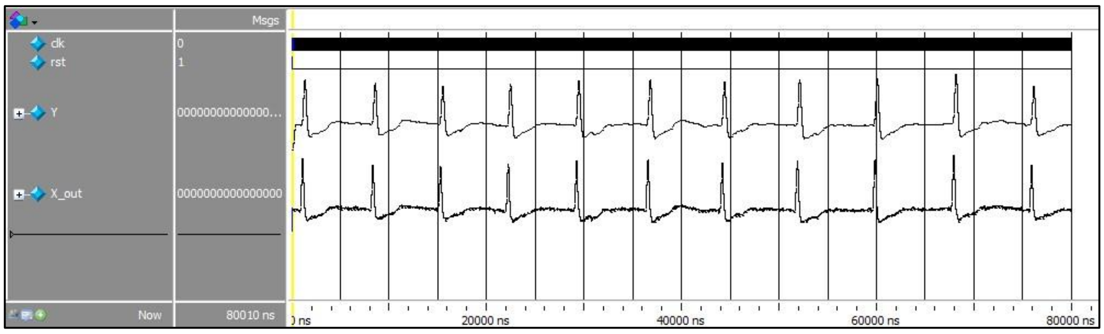


Figure 4: MATLAB outcome

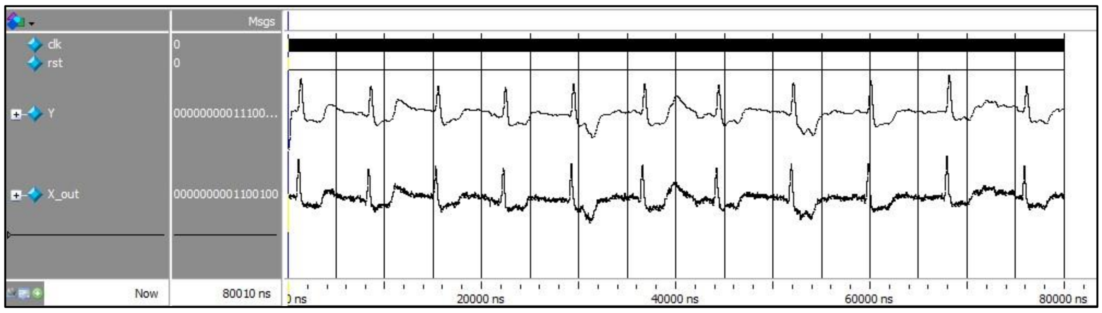
Figure 5 shows the analysis in the ModelSim Altera based on how the filter is implemented in the hardware. The X_out shows an ECG signal with noise, and the Y shows an ECG signal that has been filtered.



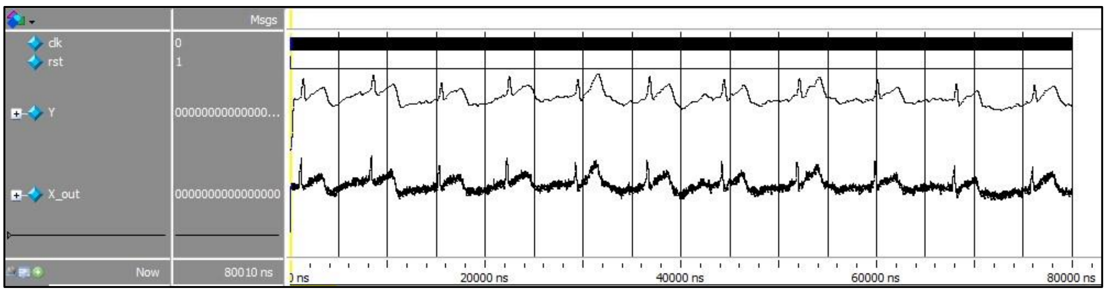
(a)



(b)



(c)



(d)

Figure 5: FPGA hardware implementation (a) ECG 1 (b) ECG 2 (c) ECG 3 (d) ECG 4

3.3 Performance Analysis

The execution time in the process of filtering the ECG signal can be used to compare the performance of MATLAB and FPGA. Table 3 displays the performance analysis between MATLAB and FPGA for all sample ECG signals. The FPGA will run in the same duration of time because each clock cycle generates output in 20 ns. Meanwhile, the running time of MATLAB is determined by the CPU performance, which can be faster or slower. MATLAB will run on an Intel Core i5 10th generation processor with 16 GB RAM and an NVIDIA GeForce GTX 1650 Ti graphics card in.

Table 3: Performance processing between MATLAB with FPGA

ECG Samples	MATLAB Running Time	FPGA Running Time	Run Time Improvement
Sample 1	20.58 ms	80 µs	257×
Sample 2	20.65 ms	80 µs	258×
Sample 3	20.81 ms	80 µs	260×
Sample 4	20.68 ms	80 µs	259×

As shown in Table 4, the running process of the FIR filter in Quartus Prime can also be analysed using the maximum operating frequency, logic utilisation, and power consumption of the hardware design. According to the table, 43% of the total logic elements are required to run the FIR filter, and the filter will use 381 registers in total. In designing the FIR filter for this project, it is evident that the cost of customizing the hardware logic will be extremely high due to the flow analysis overview in Quartus Prime simulation. This issue allows the author to reduce the use of multiplier hardware while saving costs.

Table 4: Flow analysis overview of FIR filter hardware design

Flow Analysis	MATLAB Running Time
Total Logic Element	2680
Total Registers	381
Maximum Frequency	63.81 MHz
Power Consumption	79.24 mW

4. Conclusion

The results show that the FIR filter implemented on the FPGA can process the ECG signal up to 260 times faster as compared to software implementation. This project proves that the digital signal processing on a custom design hardware logic will provide faster processing speed as compared to the software implementation due to parallel processing capability. The maximum operating frequency, logics utilization and power consumption of the hardware design were also analysed. The logics utilization and power consumption can be further reduced by implementing the operation scheduling and constrained resource allocation.

Acknowledgement

The authors would like to thank the Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia for its support.

References

[1] A. Walinjkar and J. Woods (2017), "Personalized wearable systems for real-time ECG classification and healthcare interoperability: Real-time ECG classification and FHIR

interoperability," Internet Technologies and Applications (ITA), 2017, pp. 9-14, DOI: 10.1109/ITECHA.2017.8101902

[2] M. T. Almalchy, V. Ciobanu and N. Popescu (2019), "Noise Removal from ECG Signal Based on Filtering Techniques," 22nd International Conference on Control Systems and Computer Science (CSCS), 2019, pp. 176-181, DOI: 10.1109/CSCS.2019.00037

[3] U. Satija, B. Ramkumar and M. S. Manikandan (2019), "A New Automated Signal Quality-Aware ECG Beat Classification Method for Unsupervised ECG Diagnosis Environments," in IEEE Sensors Journal, vol. 19, no. 1, pp. 277-286, 1 Jan.1, 2019, DOI: 10.1109/JSEN.2018.2877055

[4] E. Ozpolat, B. Karakaya, T. Kaya and A. Gulten (2016), "FPGA-based digital Filter Design for Biomedical Signal," 2016 XII International Conference on Perspective Technologies and Methods in MEMS Design (MEMSTECH), pp. 70-73, DOI: 10.1109/MEMSTECH.2016.7507523

[5] Zheng, J., Zhang, J., Danioko, S., Yao, H., Guo, H., & Rakovski, C. (2020). A 12-lead electrocardiogram database for arrhythmia research covering more than 10,000 patients. Scientific Data, 7(1), 1-8

[6] Li, H., & Boulanger, P. (2021). An Automatic Method to Reduce Baseline Wander and Motion Artifacts on Ambulatory Electrocardiogram Signals. *Sensors*, *21*(24), 8169

[7] Chavan, M. S., Agarwala, R. A., & Uplane, M. D. (2006, February). Use of Kaiser window for ECG processing. In *Proceedings of the 5th WSEAS Int. Conf. on Signal Processing, Robotics, and Automation, Madrid, Spain*

[8] Husain, Khaleel, Mohd S. Mohd Zahid, Shahab Ul Hassan, Sumayyah Hasbullah, and Satria Mandala. 2021. "Advances of ECG Sensors from Hardware, Software and Format Interoperability Perspectives" Electronics 10, no. 2: 105. https://doi.org/10.3390/electronics10020105

[9] Zahoor, S., & Naseem, S. (2017). Design and implementation of an efficient FIR digital filter. *Cogent Engineering*, *4*(1), 1323373.

[10] Mahawar, N., Datar, A., & Potnis, A. (2013). Performance analysis of adjustable window-based FIR filter for noisy ECG signal filtering. *International Journal of Advanced Computer Research*, *3*(3), 80