FPGA BASED IMPLEMENTATION OF A MULTIPLIER-LESS FIR FILTER FOR ECG SIGNAL PROCESSING

Foisal Ahmed[1] Md. Fokhrul Islam[2]

[1]Dept. of Electrical and Electronic Engineering, Prime University, Dhaka-1216, Bangladesh, faisal_ete@yahoo.com

[2] Dept. of Electrical and Electronic Engineering, Islamic University of Technology (IUT), Organization of Islamic Cooperation (OIC), Board Bazar, Gazipur-1704, Bangladesh.

Abstract: *This research work presents an efficient digital system using Field Programmable Gate Array (FPGA) to filter the Electrocardiogram (ECG) signal. Finite Impulse Response (FIR) Digital filter is being used to preprocessing and denoising the ECG signal. The Reduce Adder Graph (RAG) algorithm has been incorporated in implementation of the FIR filter which reduces not only the size and cost but also decreases the computational time significantly. This work has achieved the target of 50Hz noise cancellation. The output of ECG signal is then compared with the ECG signal before and after filtering by plotting the signal both in time and frequency domain using MATLAB tools. The entire system has been implemented on the ALTERA DE-II FPGA education board by synthesizing Verilog HDL using Quartus II tool*

Key words: *FPGA, RAG-n, ECG, FIR, Verilog HDL.*

1. Introduction

Reconfigurable platform such as FPGA, CPLD, and PLD etc is now being used for designing and implementing SoC due to its' low cost and high capacity and tremendous speed [1-2]. Researchers are using this platform for simulating many important algorithms in signal processing such as ECG, EEG, image processing etc in the area of Bio-medical engineering. One of the ***state of art*** FPGA which is based on look-up table structure and have the parallel execution capabilities, shows more excellent features than in the DSP in the high capacity data processing [3-4]. So, now a day it is a scope to use this FPGA device in the biomedical applications.

The Fig. 1 shows a sample of Electrocardiogram (ECG) signal. Electrocardiogram (ECG) is considered now as the important source of information which measures the physical condition of the heart. Basically, ECG is the captured of the electrical behavior of the heart generated by heart muscles on body surface. When recording this signal from the human body surface usually interfaced by some nonlinear signal such as a background and strong random noise [5].

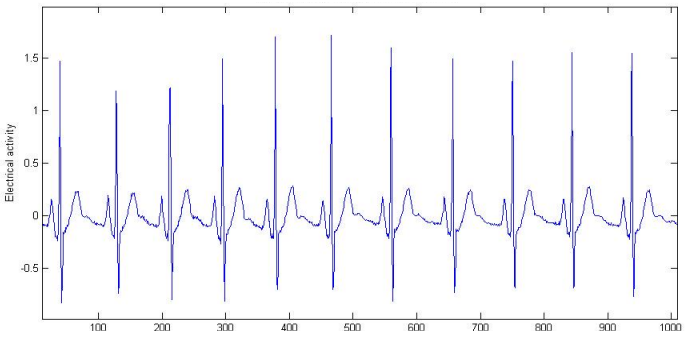

Fig. 1 Sample of ECG Signal

Therefore, the captured ECG signal is usually distorted by noise. Hence the powerline interference noise must be cancelled out for the further process like bit detection, heart rate analysis etc. There are mainly two types of noise that are observed while capturing ECG signal. One is power line interference usually 50Hz to 60Hz in frequency mainly arises from monitoring equipment and contraction noise. Every so often ECG signal is completely distorted by this powerline interference noise [6]. For the cause of human breathing some baseline wander noise also have seen which is negligible.

At present, lots of researches are doing now to remove this noise from the ECG signal.

Digital Filters play a significant role to remove this unwanted noise. Digital filters are classified as finite impulse response (FIR) and infinite impulse response (IIR) filters [7]. Although FIR filters are more complex, they have certain advantages over IIR filters, which the FIR counterparts are always stable and thus are particularly useful for applications where exact linear phase is required. Fig. 2 shows a typical block diagram of a filter. FIR filter is a one polynomial coefficient. To get an equivalent result like IIR filter, FIR filter consumes much high order polynomial, which results in longer delay.

$Y[n] = b_0x[n] + b_1x[n-1] \ldots\ldots\ldots\ldots\ldots + b_n x[n-N]$ (1)

At the right side, N is the filter order of an Nth-order filter has $(N + 1)$ terms; generally these are known as *taps in*.

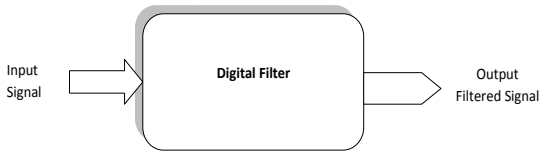

Fig. 2 Block Diagram Digital Filter

Design a FIR filter along with all of its specification is very much challenging job. Three ways can be designed the Filter such as window, sampling and optimum method. Designing FIR filters through Window method is a direct method. Window method is one of the simplest methods in designing the FIR filter because it has the ability to minimize the Gibbs' oscillations shows in the Fourier series method using window function. In this approach we truncate the Fourier series representation. [7-8].

The purpose of this work is to design a suitable low cost FIR LPF that can be used for removing 50Hz noise from ECG signal.

1.1 CONTRIBUTIONS

We make the following contributions in this work:

- Demonstration of FPGA as a FIR LPF design platform
- Low power, small area, low cost and high speed FIR Low pass Filter with all the parameters are optimized
- A Digital FIR filter which can remove 50Hz line frequency noise from ECG signal.

2. SOME RELATED WORK

The purpose of this work is to design a suitable low cost FIR LPF that can be used for removing 50Hz noise from ECG signal. For the efficiency of hardware resources in the case of cost and speed, we should optimize our filter coefficient. That's why it needs to investigate various algorithms which can reduce the size and cost. There are many algorithms which have been proposed to minimize the cost, size and power. A lot of design methods of low power digital FIR filter are proposed, for example, in paper [9], just using registers, adders and hardwired shifts they have presented a scheme to design a FIR filter. By using a elimination algorithm, they reduced the number of adders extensively. In paper [10] they have proposed a novel method for a design method for wireless data transmission of a low power digital baseband processing. Their concentration is to optimize the bit width of each filter coefficient. In [11] presents the method where they use adders and booth multipliers to reduce switching power of a FIR filter using data transition power diminution technique (DPDT).

As a complete transposed-form FIR filter comprises the multiplier block followed by a chain of delays and additions. Since adders occupy a much greater area than latches, and shifts may be hard wired, a common objective is to minimize the number of full-word adders (the "adder cost"). Another graph algorithm, [12] the n-dimensional reduced adder graph algorithm (RAG- [2]) uses a precomputed table containing, for each positive integer, all the possible graphs that can be used to multiply by that integer.

This research mainly have concentrated RAG-n algorithm to minimize hardware cost for implementing FIR filter to remove the 50Hz noise for ECG signal processing.

3. PROPOSED DESIGNED METHODOLOGY

A process for designing an FPGA-based digital filter has the main steps shown in Fig. 3. The design begins with development of performance specifications for cut-off frequency, transition band limits, in-band ripple, minimum stop band attenuation, etc. The filter is described by a C-Language specification/algorithm, which must be converted into a Verilog RTL model that can be synthesized into hardware that implements the algorithm.

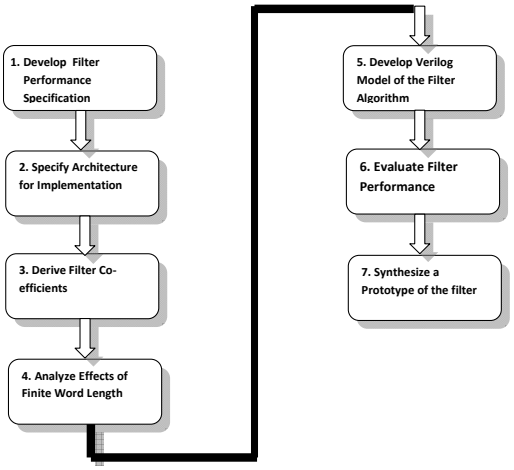

Fig. 3 Design flow for Digital Filter

The design flow is not ideal, because the algorithm's description in C must be translated into Verilog, creating the possibility for errors to occur.

Various architectures implement FIR and IIR filters [13]. For a given architecture, tools such as MATLAB can be used to determine the filter coefficients that implement a filter that satisfies the specifications of the

design. Digital filters operate on finite-word-length representations of physical (analog) values. The finite word length of the data limits the resolution and the dynamic range that can be represented by the filter, leading to quantization errors. Similarly, the representations of the numerical coefficients of the filter have a finite word length, which contributes to additional quantization and truncation error. When data are represented by integers there is an error caused by truncation of the fractional part produced by an arithmetic operation. The arithmetic operations that are performed by the filter can lead to overflow and underflow errors, which must be detected by the machine.

A. DIGITAL FILTER DESIGN APPROACH

The design specifications for a FIR digital filter include

- Characteristics of the filter (low pass, high pass, band pass)
- Pass band frequency
- Stop band frequency
- Sampling frequency
- Number of order
- How to implement the filter
- Design constraints (cost, resources limitation)

As our main objective to the application base, furthermore it will be better to discuss the characteristics of the filter in the frequency domain. We know filters are often frequency selective and for the LPF and BPF, it often to keep the specifications tightly in designing.

The parameters that are interested in filter design specification:

- Pass band ripple, $^{\delta}p$
- Stop band ripples, $^{\delta}s$
- Stop band frequency, w_p
- Pass band frequency, w_s
- Sampling frequency, f_s

To design the FIR filter as the selection of a finite sequence so that the response is much closer with the response of an ideal filter. FIR filters are generally always stable and have perfect linear phase means pure time delay and independent of phase distortion. To obtain the values of *h(n)* of the FIR filter is the objective of coefficient calculation such that the filter should meet the design specification like amplitude frequency response and throughput. There are many ways are available for obtaining $h(n)$.

This work used window based design. The reason of using this method is made of the fact that the frequency response of a filter, $H_D(\omega)$ the corresponding impulse, h_D are related by the Fourier transform [14] .

Now start with the ideal low pass response shown in Fig. 4, where *Wc* is cut off frequency and the frequency scale is normalized: T=1. By letting the response go from *-Wc* to *Wc* we simplify the integration operation. Thus the impulse response is given by:

$$H_d(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} 1^* e^{j\omega n} d\omega$$

$$= \frac{1}{2\pi}\int_{-\omega}^{\omega} e^{j\omega n} d\omega$$

$$= \frac{2f_c}{cW_c}\sin(nW_c), \text{ where } n\neq 0, -\infty \le n \le \infty$$

$$= 2f, \text{ n= 0}$$

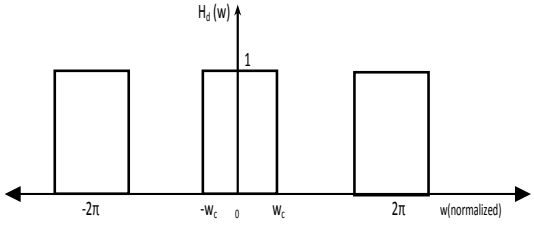

Fig. 4 Ideal Frequency response of a band pass filter

4. FIR Half Band Filter

A half-band FIR filter has an odd-length impulse response $h[n]$ whose alternate samples are zero. The main advantage of half-band filters is that their realization requires only about half the number of multipliers.

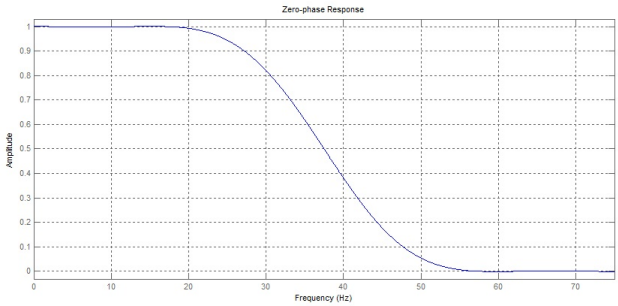

Fig. 5 Magnitude frequency response specifications for a Low-pass filter of 50 Hz Cut-off frequency

The impulse response of an ideal lowpass filter is $h[n] = 2F_C \text{ sinc}(2nF_C)$. As for example, If we choose $F_C = 0.25$, we obtain $h[n] = 2F_C \text{ sinc}(2nF_C) = 0.5 \text{ sinc}(0.5n)$, where $|n| \le 0.5 (N-1)$ and N is the filter length is always odd.

Using the Kaiser window, the magnitude frequency response of low pass filter can be obtained using the MATLAB FDA tool command as shown in Fig. 5. By asserting the desired value of stop band frequency, pass band frequency and sampling frequency for the ECG signal processing, MATLAB will generate the appropriate filter order required. For simplification, the filter coefficients are generated using the MATLAB FDA tool. For windowing the filter Kaiser Window function is used because some researches [7], it has shown better performance in designing the FIR filter.

TABLE 1
Design Specification of FIR LPF

Pass band ripple, $^{\delta}p$	.001
Stop band ripples, $^{\delta}s$	.001
Stop band frequency, w_p	60Hz
Pass band frequency,w_s	35Hz
Sampling frequency, f_s	150Hz
Order	15

Table 1 shows the design specification of FIR low pass filter in this work. This research work has used Halfband FIR filter.

5. REALIZATION OF FIR FILTER

The realization of FIR filters can be done by using the following design method [15]:

1. Choose filter structure
2. Choose arithmetic number either fixed point or floating point and number representation, e.g. signed magnitude, 2's compliment
4. Select any approach in between parallel and serial operation
5. Synthesize software code, or hardware circuit, which will carry out actual filtering operation.
6. Verify result of the simulation whether the design meets given performance specifications.

6. Process of Implementing FIR Filter

We know real ECG signal is totally analog signal in nature. On the other hand, the FIR filter is processing in the discrete form. Firstly the analog input signal must be sampled and convert binary digit using an ADC (analog-to-digital converter). The resulting binary numbers represent successive sampled values of the input signal. For doing out the numerical calculation on them, the values are then transferred to the processor. Basically, this calculation is a convolution process. These calculations usually involve multiplying the input samples by constants (coefficients) and summing the products together. Sometimes, the output result of the filtered signal, are output through a DAC (digital-to-analog converter) to convert the signal back to analog form [15].

6.1 Realization of Filter Structure

For a given transfer function H(z), it is often important to choose a specific filter structure. In the design of fixed point, the choice of digital filters is usually based on reducing length of finite register lengths.

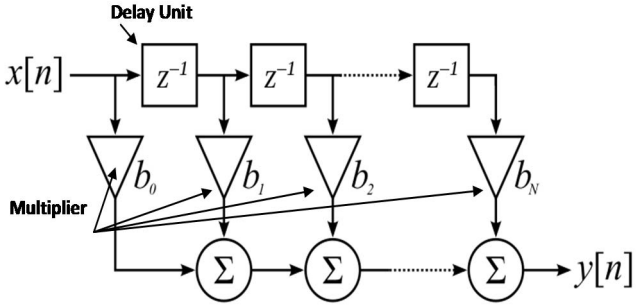

Fig. 6 Structure of Direct form of FIR filter of length N

There are direct-form structures of FIR Filter. These direct structures are effected by coefficient sensitivity problems, which means, for large value of the order of filter the poles (in case of recursive filters) and zeroes locations could be changed. In our research, Direct form of FIR filter has been implemented whose new look is given in Fig. 6. x[n] is the input data, y [n] is the output data and b_0, b_1,b_2 and b_N are the Filter co-efficient.

6.2 Data Representation of coefficients

In general, there are two kinds of Data representation, one is fixed-point representation, and the other is IEEE floating-point representation.

In this work, the procedure of representing the filter coefficients and input samples is given as below:

A binary point is usually set between the first and second bit positions of the register as shown in Fig. 7 is as given below [16]. The addition or subtraction of two fixed-point numbers falling in a given range may produce a result outside that range, though. Such a result, called overflow, it must be either avoided, or corrected during DSP calculations. We are avoiding here. The data is represented in fixed-point notation. In the fixed-point format, the numbers are usually assumed proper fraction.

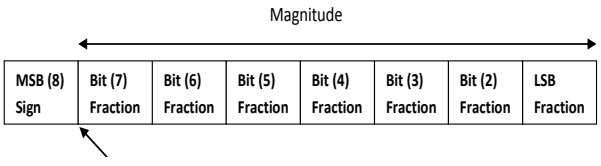

Fig. 7 Numeric Representation Format in 8-bit

To design a real filter, it requires an infinite word length to represent the filter coefficients. However, it is not possible to do so. Because the FPGA device have finite number of memory and limited hardware resources. Therefore, an appropriate approach to solve this problem is truncated the filter coefficients to an X-bit representation. In order to minimize the hardware used in this research work, the filter coefficient will be quantized to 8-bit data representation for simplicity of prototype design. Now, the filter coefficient has an 8-bit data length whereby the most significant bit represents the sign of the data either it is positive or negative. The remained 8-bit is used for the magnitude of fractions. In fig. 7, it shows the numeric representation format of 8-bit data width.

Since the coefficient number is in floating point format, so before processing, it has to be converted to integer. We know, MATLAB can work with very high precision number but in real life it is not possible to design a filter using such coefficients number. So that this number has been truncated into seven numbers after decimal point and converted into 8-bit integer number shown in Table 2.

TABLE 2
Comparison between Original and modified Coefficients

Tap	Original	Truncated	Integer
1	-0.000927100290813923	0.0009271	0
2	0	0	0
3	0.0141284232418152	0.0141284	3
4	0	0	0
5	-0.0648900406874813	-0.0648900	-16
6	0	0	0
7	0.302018027103051	0.3020180	77
8	0.500000000000000	0.5000000	128
9	0.302018027103051	0.3020180	77
10	0	0	0
11	-0.0648900406874813	-0.0648900	-16
12	0	0	0
13	0.0141284232418152	0.0141284	3
14	0	0	0
15	-0.000927100290813923	-0.0009271	0

7. Hardware Implementation of FIR Filter

By using only an adder, a multiplier and registers we can implement an FIR filter easily. For optimization purpose, the number of multipliers should be limited as for the hardware resources used. If we are using a lot of multipliers the system is not area-efficient. For hardware implementation, if a designed filter has 15 orders, it requires 16 multipliers for the computation. Implementing 16 multipliers in FPGA board is not an appropriate way as it is not area-efficiency.

To implement a multiplier, we need so many adders which made the system complexity. In this research work we mainly focused to use dependence graphs of the coefficients to reduce the number of operation for implementing the multipliers. That's we need the help Graph dependence algorithms to reduce the number of adders in implementing the multipliers. In the embedded system design speed and power consumption always have been challenging job. Lots of research has been done to obtain this goal as we mentioned previously ([8 -12]).

The n-dimensional reduced adder graph algorithm (RAG- [2]) uses a precomputed table containing, for each positive integer, all the possible graphs that can be used to multiply by that integer [12]. Therefore, a RAG-[2] technique is proposed in this research work to implement the hardware of FIR digital filter for ECG signal processing.

7.1 Reduced Adder Graph of N Dimension (RAG-n)

The RAG-n algorithm consists of two segments. The first segment is a precise algorithm and the second part is a try and error method. First of all, the minimum adder cost is possible if the coefficients are completely synthesized. We build up a look-up table for each coefficient being used in the second part. The algorithm essentially consists of the following steps:

- Reduce all coefficients in the set to odd fundamentals.
- Using the cost lookup table evaluate all single-coefficient costs.
- Remove all cost-0 fundamentals.
- Make the graph representation of elected fundamentals.
- Look at pair wise sums of fundamentals in the graph set multiple with power-of-2 of these same fundamentals.
- Do again step 5 so that no more fundamental remains.

7.2 Reduced Adder Graph (RAG-n) for Proposed Half-Band FIR Filter

For our proposed halfband FIR filter has seven nonzero coefficients, namely b[3], b[5], b[7], b[8], b[9], b[11], and b[13] which are 3, -16, 77, 128, 77, -16 and 3. For a first cost estimation we change the decimal values into binary representations for simplicity. Then search co-efficients for the cost minimization using Table given in the RAG algorithm [12]. It follows in the Table 3:

TABLE 3
Cost Estimation for the Coefficients using RAG-n

Step	To be realized	Already realized	Action
1.	{3, -16, 77, 128}	{-}	Initialization
2.	{3, 16, 77, 128}	{-}	Remove Negative Sign
3.	{3, 77, }	{16, 128}	Remove 2^k Co-efficient
4.	{3, 77, }	{-}	Remove 2^k factors from Co-efficient
5.	{77}	{3 }	Realize cost 1 Co-efficient
6.	{77}	{3 }	Other coefficient is prime number
Apply the heuristic to the other co-efficient, starting with the lowest cost and small value			
7	{77}	{3 }	For 3 = 2 + 1
8	{-}	{3, 77, }	For 77 = 64 + 13 = 64 + (8+4) +1

Fig. 8 shows the resulting Reduced Adder graph. The number of adder is reduced from 4 to 3. The adder path delay also reduced from 4 to 2.

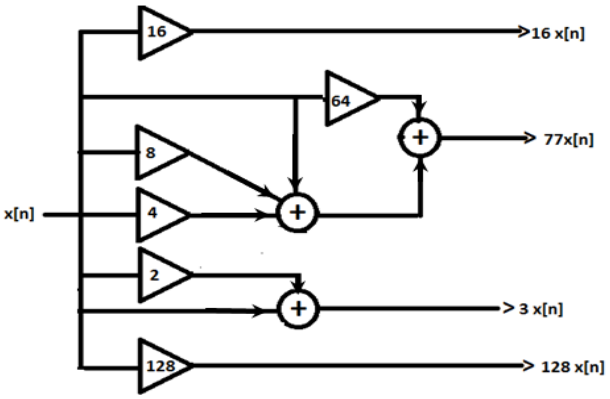

Fig. 8 Realization of the coefficients using RAG-n Algorithm

8. Results

The ultimate objective of this research work is to get the FIR low pass filter for ECG signal processing to remove 50Hz on the Altera DE2 Board. In order to achieve this objective, FIR low pass filter design is divided into two main parts. Firstly, the filter is initially designed and simulate using MATLAB simulator. The ECG signals are filtered using MATLAB simulator to test the functionality of the filter. The filter coefficients are to be quantized and rounded off through MATLAB.

For the design purpose, a register-transfer level (RTL) design method is used for implementing this filter on the FPGA. RAG algorithm is used to minimize the hardware cost. The design of the filter is developed by Verilog HDL code. The designed file is then analysis and synthesized using Quartus-II tools and for the simulation purpose the Vector Wave From (VWF) tool is used. After successfully implementation, the proposed designed is compared with the again MATLAB simulator for the validity.

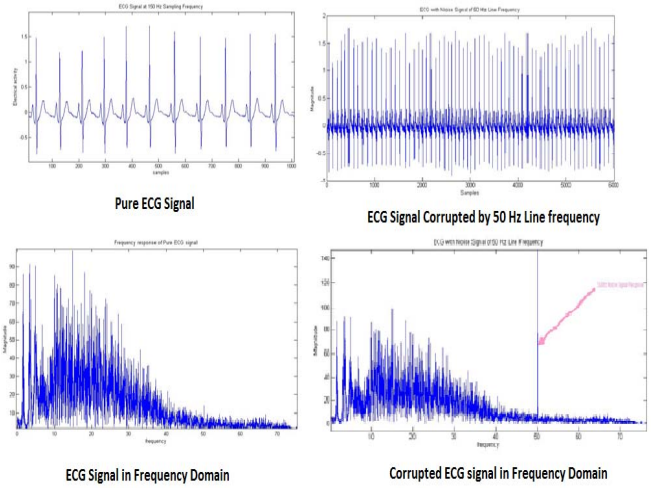

Fig. 9 Pure and Corrupted ECG signal in Time and Frequency domain.

In Fig. 9 the ECG signal is plotted in both time domain and frequency domain. This ECG sample data has been collected from the **MIT-BIH** Arrhythmia database [17]. The data were sampled at 150Hz. It is easier to analyze noise region of a signal by plotting it in the frequency domain. The designed filter is then tested by feeding the corrupted ECG signal. We have seen this FIR filter totally eliminates the 50Hz noise from the ECG signal successfully. The Fig. 10 shows the total operation at a glance through half-band FIR filter by using MATLAB tools.

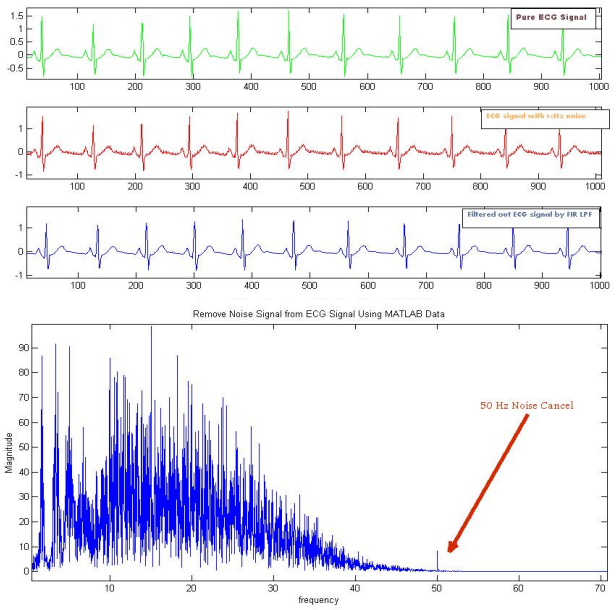

Fig. 10 Filter out 50Hz Noise signal from ECG.

Now, this digital FIR LPF system has been tested using the Quartus-II tool. The filter coefficients are directly used in Verilog HDL Code to develop the filter designed. The RTL view of the FIR filter is shown in Fig. 11.

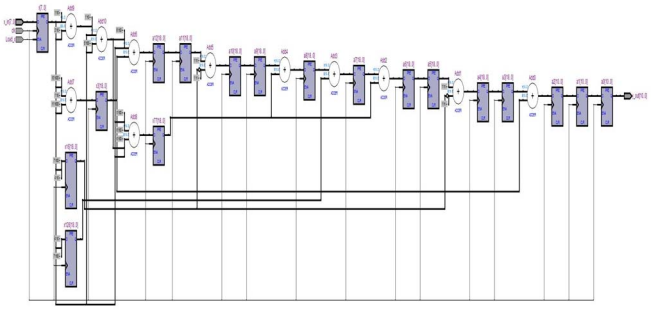

Fig. 11 RTL view of FIR LPF

The Fig. 12 shows the VWF simulation file using Quartus-II tools for some discrete inputs. The output is verified with MATLAB simulator and found the similar result. So our designed FIR filter would successfully remove the noise from ECG signal.

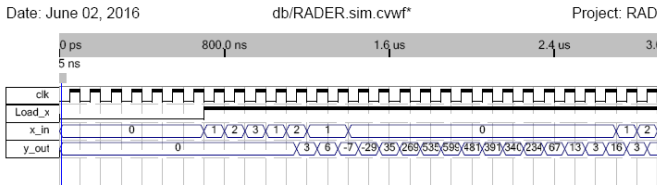

Fig. 12 VWF Simulation result of FIR filter

9. Resource Utilization

The proposed filter has been implemented in the ALTERA DE II FPGA board and device family name Cyclone II, model no: EP2C5F256C8. The Hardware resources are summarized in Table 4

TABLE 4
Resources Utilization of the Hardware of the proposed designed

Resources	Used	Percentages (%)
Total Logic elements	125 out of 4,608	3%
Total combinational functions	93 out of 4,608	2%
Dedicated logic registers	110 out of 4,608	2%
Total pins	21 out of 158	13%
Total memory bits	0 out of 119,808	0%
Embedded Multiplier 9-bit elements	0 out of 26	0%

To compare the proposed designed we have used ordinary filter which have architecture of without using RAG algorithm on the same FPGA device. In the ordinary filter the data is directly convoluted with the filter coefficients. In that case multiplier is used for that operation. Consequently, in the ordinary filter realized more hardware resources and consumed more power. Hardware resources of this ordinary filter are summarized in Table 5.

TABLE 5
Resources Utilization of the Hardware of the Ordinary designed

Resources	Used	Percentages (%)
Total Logic elements	304 out of 4,608	7%
Total combinational functions	114 out of 4,608	2%
Dedicated logic registers	304 out of 4,608	7%
Total pins	39 out of 158	25%
Total memory bits	0 out of 119,808	0%
Embedded Multiplier 9-bit elements	6 out of 26	23%

Fig. 13 shows the performance chart of the proposed designed using RAG and the ordinary filter.

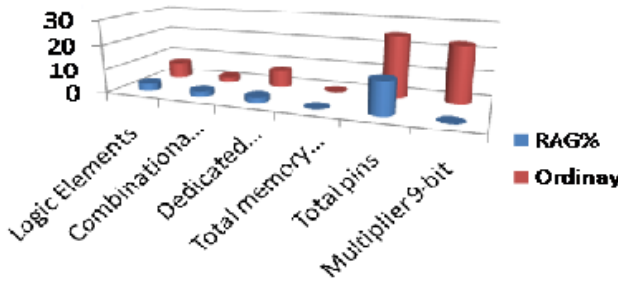

Fig. 13 Comparison Chart of resources between proposed and ordinary filter

We have also analysed the power consumption of our proposed filter and ordinary filter. The Fig. 14 shows the comparison chart of the power summery.

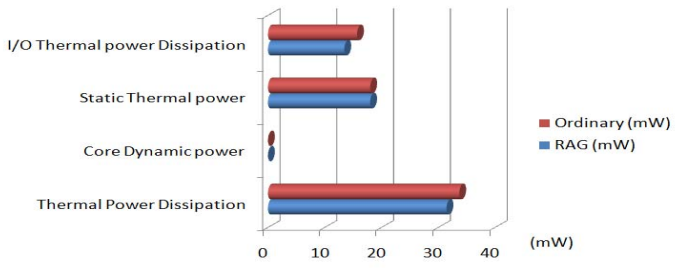

Fig. 14 Chart of power consumption summery between Proposed and ordinary design.

Therefore, this proposed designed FIR filter not only removes the 50Hz noise signal from the input data ECG signal but also reduces the size, power consumption and cost. This work also compares the result with the MATLAB outcome and makes the feasibility analysis for real life implementation. MATLAB simulator also used to measure the coefficients of the designed filter to analysis the ECG signal.

10. Conclusions And Recommendation

The aims of this research work is to studied, analysed and implemented FIR low pass digital filter to remove 50Hz noise in the electrocardiogram (ECG) signal. The desired low pass digital filter is implemented using Verilog HDL code on Altera FPGA DE2 board. Implementation of FIR filters on FPGA is essential because it can enhance the speed. In terms of high speed architecture, the direct form approach is preferred for design and RAG-n algorithm has been incorporated to design the architecture and successfully implemented in FPGA before investigated it in the MATLAB simulator tools. Fabrication time to marketplace, cost effective for small production volume and reconfigure ability make FPGA devices an ideal solution for many biomedical, military and university researches. The design implementation entailed the employment of Altera Quartus-II software tool. Implementing the design on a Cyclone-II chip of EP2C5F256C8 and hardware testing and verification of the filter has been done by MATLAB tools. Finally simulation and synthesis with of Altera Quartus-II tool and RTL schematic of filter chip obtained.

This proposed synthesized multiplier-less hardware design for the ECG signal processing system indicates that usage of the hardware resources such logic elements of the FPGA is small enough to be implemented in currently available parts. This is an important aspect which makes the system feasible to be ultimately fabricated as a complete hardware system or embedded within a handy small electronic device such as a smart-phone where patient can observe their pure ECG signal.

References

1. Ali, L., Sidek, R. I., Aris, M. A., Ali, M. and Suparjo, B. S.: (Challenges *and Directions for IC testing).* In: Integration, the VLSI Journal, Elsevier Science, Netherland, 2004, Vol. 37(1), p. 17-28.
2. Obermaisser, R., Kopetz, H. and Paukovits, C.: (*A Cross-Domain Multiprocessor System-on-a-Chip for Embedded Real-Time Systems).* In: IEEE Transaction on Industrial Informatics, Vol. 6(4), 2010.
3. Song, P., Jiye, H. and Guodong, W.: (*Modern DSP Technology [M]).* In: Xi'an: Xidian University Press, 2003, China, p. 163-180.
4. Xiongfei, L., Jinding, G. and Haibing, Q.: (*A new way on FPGA implementation of LMS adaptive filter).* In: Piezoelectrics & Acoustooptics, 2007, 29(1): p.87-89.
5. Abedin, Z. and Conner, R.: (*ECG interpretation The self assessment approach).* Blackwell Publishing, 2008.
6. Hampton, J. R.: (*The ECG in Practice).* Elsevier Health Sciences, 2008.
7. Rajput, S. S. and Bhadauria, S.S.: (*Implementation of FIR Filter using Adjustable Window Function and Its Application in Speech Signal Processing).* In: International Journal of Advances in Electrical and Electronics Engineering, Vol. I, p. 158-164.
8. Ketha, M. B., Venkateswarlu Ch. and Raghuram, K.: *(Design & Fpga Implementation of Reconfigurable Fir Filter Architecture For Dsp Applications).* In: International Journal of Engineering Research & Technology, Vol. I, Issue 7, September 2012.
9. Mirzaei, S., Hosangadi, A. and Kastner, R.: (*FPGA Implementation of High Speed FIR Filters Using Add and Shift Method).* In: Proceedings of the International Conference of Computer Design, IEEE, 2006.
10. TARUMI, K., HYODO, A. and MUROYAMA, M. and YASUURA, H.: (*"A design method for a low power digital FIR Filter indigital wireless communication systems).* In: Catalog of Library, System LSI research center, Kyushu Univresity, 2004.
11. Senthilkumar, A. and Natarajan, A. M.: (*Design and Implementation of Low Power Digital FIR Filters relying on Data Transition Power Diminution Technique).* In: DSP Journal, Vol. 8, p. 21-29, 2008.
12. Dempster, A. G. and Macleod, M. D.: (*Use of minimum-adder multiplier blocks in FIR digital filters).* In: *IEEE Trans. Circuits Syst. II, Analog. Digit. Signal Process.* vol. 42, no. 9, p. 569–577, Sep. 1995.
13. Ciletti, M. D.: (*Advanced Digital Design with the Verilog HDL).* 1st ed., Prentice-Hall. Inc. of India, 2003.
14. Proakis, J.G., Manolakis, D.G.: (*Digital Signal Processin.* 3rd ed, PHI publication, 2004.
15. Antoniou, A.: (Digital *Filter).* 3rd ed, Tata Mc. Graw Hill publications, 2001.
16. Jones, D. L.: (*FIR Filter Structures).* Version 1.2: Oct 10, 2004.
17. Physio NET: *www.physionet.org*
18. Nian-qiang, L. H., Si-Yu and Shi-yao, C.: (*Application of Distributed FIR filter based on FPGA in the analyzing of ECG signal).* In: Proceedings of the IEEE International Conferenece on Intelligent System Design and Engineering Application, 2010.
19. Singhal, E.: (*Performance Analysis of Finite Impulse Response (FIR) filter Design Using Various Window Methods).* In: International Journal of Scientific Research Engineering & Technology (IJSRET), vol. 1, issue 5, p. 18-21, Aug 2012.
20. Bhaskar, P. C. and Uplane, M. D.: (*FPGA based digital FIR multilevel filtering for ECG denoising).* In: Proceedings of the International Conference on Information Processing (ICIP), Dec 16-19, 2015.