



FPGA Implementation of Efficient Adaptive Filter Incorporating Systolic Architecture



Harith H. Thannoon*, Ivan A. Hashim

Electrical Engineering Dept., University of Technology-Iraq, Alsina'a street, 10066 Baghdad, Iraq.

*Corresponding author Email: eee.20.01@grad.uotechnology.edu.iq

HIGHLIGHTS

- This paper presents the refinement of systolic-based adaptive filters to mitigate critical path delay.
- The impact of utilizing a systolic architecture on filtering performance was evaluated.
- LMS, RLS, and AP algorithms were implemented within an FPGA platform.

ABSTRACT

This paper introduces an approach that capitalizes on the retimed delay concept to enhance adaptive filters' operational efficiency. In particular, it introduces an adaptive filter configuration with minimal critical path delay. The adaptive filters, namely Least Mean Square (LMS), Recursive Least Square (RLS), and Affine Projection (AP) have been meticulously designed through the utilization of the Xilinx System Generator (XSG). The approach is based on a systolic architecture, which aims to reduce the critical path delay by minimizing the number of processing elements, including adders and multipliers, in each iterative process. Notably, the critical path of the proposed filters has been successfully reduced to a single multiplier. Moreover, the non-restoring division algorithm has been employed to execute division operations within the FPGA for the purposes of weight updates in the equations of RLS and AP filters. The performance of the proposed filters was evaluated using many filter design metrics, including SNR, power consumption, steady-state MSE, convergence speed, and complexity. The improvement in SNR was 4.3%, 8.2%, and 10% for RDLMS, RDRLS, and RDAP filters, respectively. The power consumption was reduced by about 40.5%, 28.6%, and 5.9% for RDLMS, RDRLS, and RDAP filters. Moreover, the results show significant improvement in the convergence speed. The proposed filters can efficiently remove PLI noise from ECG signals with high speed and low power consumption at the cost increase in complexity, but they are still implementable on the FPGA platform. The proposed filters were implemented using the Spartan-6 xc6slx16-2csg324 FPGA.

ARTICLE INFO

Handling editor: Jawad K. Ali

Keywords:

Adaptive filter
Systolic
FPGA
non-restoring
LMS

1. Introduction

Adaptive filters are used in many applications, including noise cancellations [1], channel equalization [2], and system identification [3]. In the context of noise cancellation, adaptive filters include two parts: filtering and adaptation. Filtering involves passing the corrupted signal through the filter structure to remove unwanted signals. To perform an adaptation operation, the filter coefficients are updated to minimize the difference between the input signal and the desired signal [4]. The main operation of these filters is minimizing the error square between the desired signal and the input signal. Various adaptive algorithms are used to minimize this error, such as Least Mean Square [5], Recursive Least Square [6], and Affine Projection [7], which have been proven robust approaches for noise removal applications. Moreover, these algorithms have many advantages and limitations, making them suitable for many applications. Corrupted ECG signals are particularly improved by using adaptive filters, which can provide a higher level of accuracy and efficiency that other approaches cannot achieve.

Utilization of systolic architecture in adaptive filter design improves the performance of these filters in terms of SNR, convergence speed, and steady-state mean square error. This architecture uses a parallel process to perform filtering tasks. The process elements of the filter work separately from each other and cooperate with others to generate the output [8]. An adaptive filter designed with systolic architecture can adapt efficiently to input and desired signal change and give accurate output. Also, by utilizing this architecture, achieving optimal filtering capability. Moreover, a reduction in power consumption of these filters can be achieved by using this architecture because of parallel processing [9].

The rest of this paper is organized as follows: Section 2 discusses some previous works. Section 3 reviews the mathematical theory of adaptive filter algorithms. Section 4 systolic architecture. Section 5 presents the proposed filter structure. Simulation Results and hardware utilization are presented in Section 6. Section 7 concludes the paper.

2. Related work

The literature presents numerous approaches to enhance ECG signals using adaptive filters. Gomathi et al. [10] developed an adaptive filter by incorporating systolic and folded architectures using the XSG tool. This innovative method aims to increase convergence speed and reduce complexity. However, their study involved designing the RLS algorithm outside of the XSG tool, specifically the division term, and only used SNR and complexity metrics to validate their design. A combination of folded and systolic architectures speeds up processing process times and reduces the space needed. RLS is a highly complex algorithm and implementing it outside of the XSG tool may not provide true FPGA implementation, potentially resulting in different performance than intended. Nevertheless, using SNR and complexity metrics for design validation is valid. Further research is required to comprehensively evaluate the performance of this filter using additional design metrics to validate the filter's design. Jayapravintha et al. [11] researched various adaptive filters that incorporate systolic architecture to increase convergence speed and improve overall performance. It is worth mentioning, in their study, that the RLS and AP algorithms were designed independently of the Xilinx System Generator XSG tool. To measure the effectiveness of their proposed filter, they employed only SNR as a metric for evaluating the filter performance. The research conducted by Jayapravintha et al. [11] underscores the significance of incorporating systolic architecture with adaptive filter algorithms to enhance filter performance. Notably, the RLS and AP algorithms were designed independently of the Xilinx System Generator XSG tool, highlighting the challenges of performing division in the FPGA platform. Mahesh et al. [12] utilized a systolic architecture to create an adaptive algorithm based on LMS. They employed three distinct filter structures using the LMS algorithm. However, the proposed filter was found to have a low convergence speed, which is a significant drawback. Additionally, the validation process only considered SNR and complexity, which may limit the overall effectiveness of the filter. V. Kavitha and colleagues [13] utilized a gate-level modification architecture to improve the SNR of the adaptive filter further. They also designed the division term of RLS outside of the XSG environment. The validity of their proposed filter was tested using both SNR and complexity metrics.

Authors in [14 -17] employed various architectures to enhance the performance of adaptive filters. However, none of these studies addressed the XSG implementation of the RLS and AP algorithms. While their proposed filters were analyzed regarding SNR, complexity, and Steady-state Error, most did not include measures of convergence speed, mean square error (MSE), and power consumption for comparison. Furthermore, the tradeoff between filter characteristics is a significant drawback of the adaptive algorithms reported in the literature. Attempts to increase the convergence speed of the filter by adjusting its parameters result in an increase in steady-state MSE, which ultimately reduces the filtering capability. Most previous systolic-based adaptive filters lack comprehensive analysis, and implementing RLS and AP filters on an FPGA outside of the XSG tool may not provide true FPGA performance. Further research is necessary to thoroughly assess the performance of systolic-based filters using a broader set of design metrics to validate filter design. Additionally, this work presents a real implementation of RLS and AP by utilizing a non-restoring division algorithm.

To address the aforementioned issues, further research is required to develop adaptive filters that can enhance convergence speed without compromising filtering capability. This research involves comprehensively analyzing SNR, complexity, steady-state error, convergence speed, MSE, and power consumption. Furthermore, both the RLS and AP algorithms were designed in the XSG environment, utilizing a non-restoring division algorithm. The proposed design is implemented on the FPGA platform, benefiting from several advantages, including low design costs, high speed, and flexibility [18]. The use of systolic architecture and non-restoring division in the RLS and AP adaptive filters enhances the overall performance of the filters, rendering them more efficient and effective solutions in noise cancellation applications. The primary objective of this paper is to develop adaptive filters with efficient filtering capabilities by employing a modified version of the systolic filter structure, which minimizes the critical path delay to the time required by a single multiplier. The proposed filters aim to address the challenges of reducing tradeoffs between the filter parameters and achieving a balance between convergence speed, SNR, and power consumption through the use of systolic architecture. Moreover, another objective is to implement the designed filter on the FPGA platform and attain a highly efficient and accurate implementation of the division term by utilizing the non-restoring division algorithm.

3. Adaptive filter algorithms

The noise canceller adaptive filter is a mathematical algorithm used to reduce the total power at the output of the noise canceller. This process involves maximizing the SNR and minimizing the MSE of the output signal. Many adaptive algorithms have been developed and implemented in this study, which is described in this section. By utilizing these adaptive algorithms, the quality of the ECG signals can be improved, and enhance its usefulness for a wide range of applications [6].

3.1 Recursive least squares (RLS) algorithm

The Recursive Least Squares (RLS) algorithm is an adaptive algorithm that recursively updates the coefficients of a filter while minimizing the Mean Square Error (MSE). New samples are processed recursively at each iteration, resulting in a faster convergence speed compared to other algorithms. The effectiveness of RLS is particularly in non-stationary environments. The algorithm is presented below as a set of equations [19].

$$e(n) = d(n) - y(n) = d(n) - x^T(n) * w(n) \quad (1)$$

$$w(n+1) = w(n) + e(n) * g(n) \quad (2)$$

$$g(n) = P(n) * x(n) \{ \lambda + x^T(n) * P(n) * x(n) \}^{-1} \quad (3)$$

$$P(n+1) = \lambda^{-1} * P(n) - g(n) * x^T(n) * \lambda^{-1} * P(n) \quad (4)$$

In these equations, the output of the adaptive filter is denoted by $y(n)$, which represents the convolution in the time domain between the input signal $x(n)$, and the weights of the filter are denoted by $w(n)$ vector. The difference between the desired signal $d(n)$ and the filter output signal given the error signal $e(n)$. The filter's performance is determined by the forgetting factor λ which affects both convergence speed and signal-to-noise ratio (SNR). Additionally, the gain vector $g(n)$ plays an important role in the overall effectiveness of the filter. During each iteration, the cross-correlation matrix $P(n)$ undergoes an update equation [20].

3.2 Least mean squares (LMS) algorithm

The LMS algorithm is the most widely used adaptation algorithm in the field of noise cancellation. The simplicity of its construction contributes to its popularity, which is derived based on the least square criterion [21]. The stochastic gradient descent method is utilized in this algorithm, where the weights are updated at each iteration of the filter structure according to the least square criterion. Three fundamental equations comprise the LMS filter: the output equation, the estimated error equation, and the weight update equation [22].

$$y(n) = d(n) - x^T(n) * w(n) \quad (5)$$

$$e(n) = d(n) - y(n) \quad (6)$$

$$w(n+1) = w(n) + 2 * \mu * x(n) * e(n) \quad (7)$$

The tap step size is denoted by μ . The new weight vector is represented by $w(n)$, while $x(n)$ refers to the input signal that is being filtered. The error signal is denoted by $e(n)$, representing the difference between the filtered signal and the desired output. The new weights can be calculated from the current weights and error signal. The amounts of weights updated at each iteration are determined by the step size parameter.

This step size parameter must be chosen carefully to achieve a balanced level between convergence speed and SNR. In addition to step size parameter selection, the weight update equation of this algorithm must be designed to minimize the error signal at filter output [23]. These designed criteria are very important in designing LMS filters.

3.3 Affine projection (AP) algorithm

The AP is an effective algorithm for improving corrupted signals. In the filtering concept, the main purpose of the AP algorithm is to estimate the filter coefficients that give minimum output error using a set of input-desire samples. This algorithm can easily remove the time-varying noise, which is a challenge with traditional filters. The following fundamental equations can express the AP filter [24].

$$y(n) = d(n) - x^T(n) * w(n) \quad (8)$$

$$e(n) = d(n) - y(n) \quad (9)$$

$$w(n+1) = w(n) + (\mu * x(n)) [x^T(n) * x(n) + \delta I]^{-1} * e(n) \quad (10)$$

In this context, the output error of the noise canceller is denoted by $e(n)$. The desired signal is represented by $d(n)$, while the weight vector of the filter is denoted by $w(n)$. Additionally, the input signal is $x(n)$, the step size parameter is μ , and the regularization parameter is denoted by δ . The regularization parameter δ and the step size parameter μ are very important in this process because they govern the filter performance. The value of δ helps to prevent overfitting and improve the model's generalization. Meanwhile, the μ determines the rate at which the filter reaches its steady state, which can significantly impact the speed and accuracy of the learning process [25].

4. Systolic architecture

Systolic architecture is a type of computer architecture that performs parallel processing efficiently. The name of this architecture was given from the use of systolic arrays of processing elements that are regularly arranged and operate in a synchronized manner. Data processing, including signal processing and processes requiring high speed, such as scientific simulation, are suited to the systolic architecture [26]. It also has widespread applications in noise canceling and improving signal efficiency. It is also used in machine learning and artificial intelligence, where it can speed up the training phase of neural networks. Systolic architecture is a significant advancement in computer architecture, offering an effective method for parallel processing [27].

Systolic architecture is an advanced technology for filter design that enables efficient real-time data processing by utilizing a specialized architecture. By incorporating systolic architecture into filter design, the resulting filters exhibit high efficacy in various applications. To implement this architecture, delay elements are inserted between the filter's process element, including adders and multipliers. This modified version of the adaptive filter is denoted as a retimed delay-based adaptive filter. By inserting a delay in the weight update and error equations of the LMS filter, for example, the RDLMS filter (LMS filter design using systolic architecture) can be achieved. To express the weight update and error equations of the LMS filter after the delay has been inserted, the following equations can be utilized [8]:

$$w(n+1) = w(n) + 2 * \mu * x(n-m) * e(n-m) \quad (11)$$

$$e(n-m) = d(n-m) - y(n-m)e(n) = d(n) - y(n) \quad (12)$$

Where the inserted delay is represented by the m , μ is known as the step size, the input vector is represented by $x(n)$, and $w(n)$ is the filter coefficient vector. By inserting the delay elements into the filter structure, the systolic architecture can be effectively applied, this modification enables the efficient processing of data and enhances the performance of the adaptive filter.

The normal configuration of a 4-tap FIR adaptive filter is shown in Figure 1, where the input signal is multiplied directly by the filter coefficients. The red line indicates the critical path delay of this structure, which contains three adders and one multiplier. To minimize this delay, Systolic architecture is applied to modify the structure of the adaptive filter, as illustrated in Figure 2 [10]. This new filter structure reduces the critical path delay to one multiplier and one adder, improving overall performance.

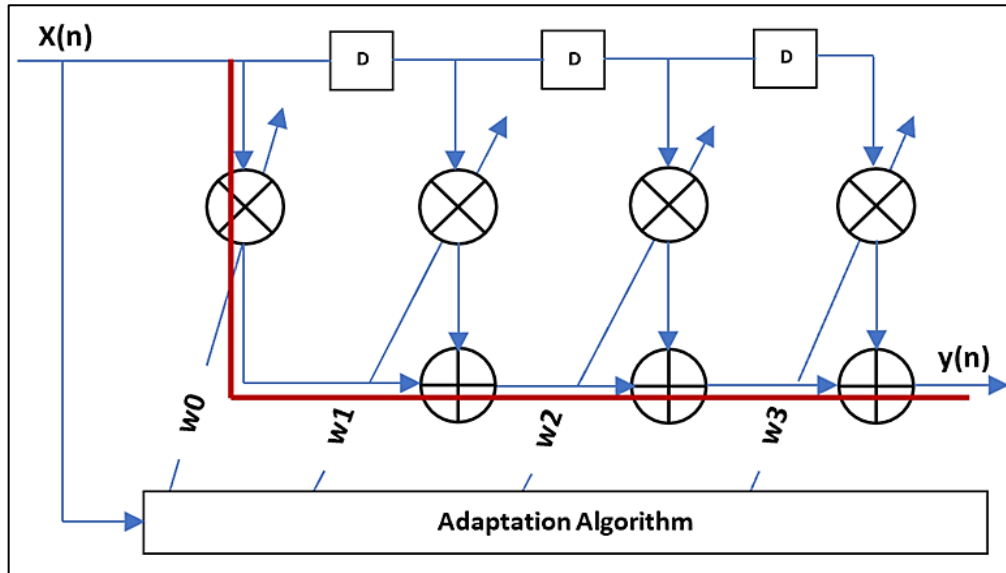


Figure 1: Conventional 4-tap Finite Impulse Response (FIR) adaptive filter without systolic architecture

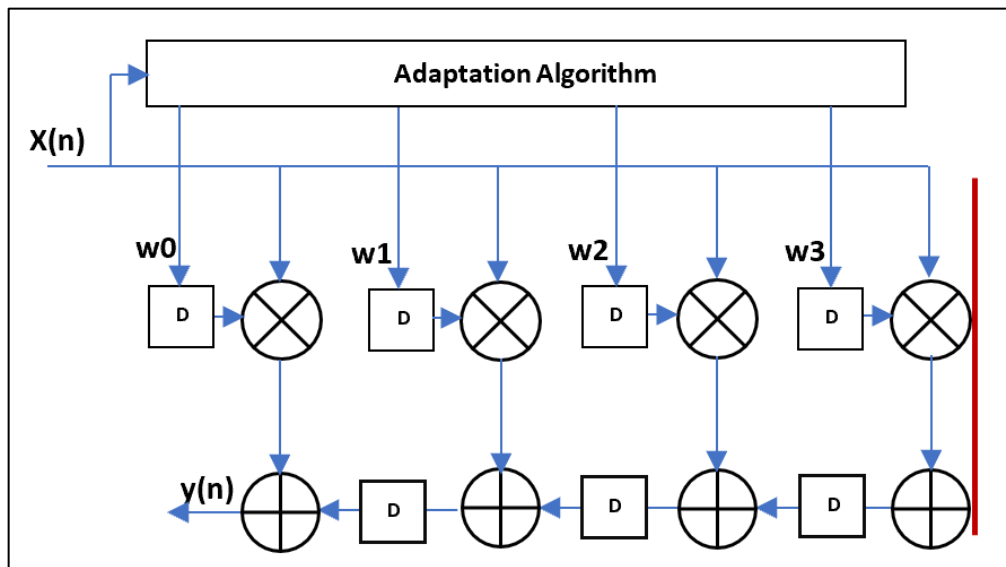


Figure 2: 4-tap Finite Impulse Response (FIR) adaptive filter with systolic architecture

5. Proposed retimed delay filter structure

The proposed systolic architecture has undergone additional refinements, wherein a modification to the systolic-based filter was introduced. This modification entails the insertion of delay blocks between the various processing elements, such as adders and multipliers, within each filter tap. This modification aims to attain a minimized critical path delay, ultimately consolidating it to just one multiplier. This modified version has proven to be highly effective in enhancing the performance of adaptive filters compared to the previous iterations of systolic-based designs. The principles of retimed delay were specifically applied to optimize the LMS filter, as depicted in Figure 3. In this representation, the "D" block symbolizes a delay operation, while "2D," "3D," and "4D" denote two, three, and four delay operations, respectively, which have been inserted to prevent misalignment between filter samples when implementing the systolic architecture. Notably, the red line in the figure represents the critical path delay of the LMS filter, which, in this instance, comprises only one multiplier. It is important to emphasize that the insertion of delays does not alter the output equation of the filter.

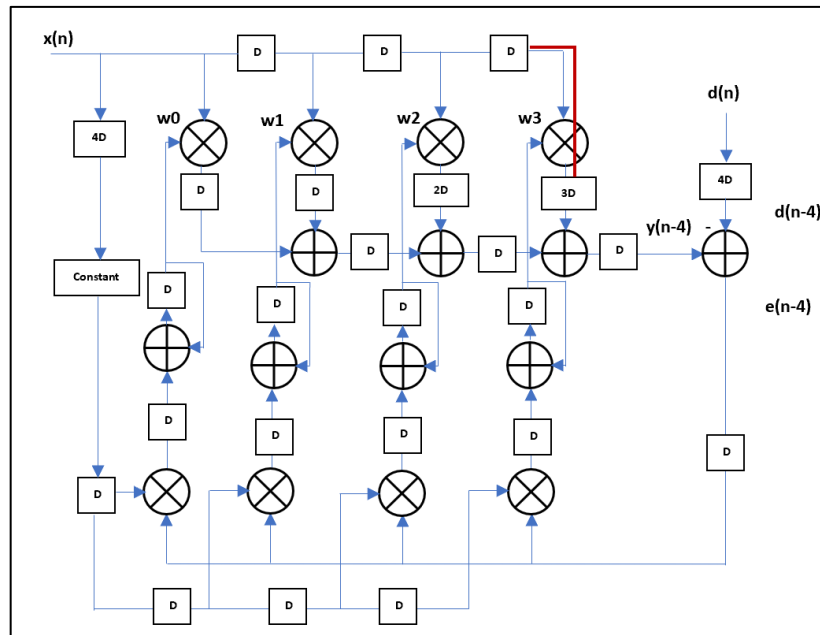


Figure 3: Proposed 4-tap LMS adaptive filter with retimed delay principle

6. Results and discussion

This research discussed the design and implementation of 4-tap LMS, RLS, and AP filters, both with and without the utilization of systolic architecture. XSG in ISE 14.7 and MATLAB 2012a were used to design these filters, which were subsequently implemented on the Spartan6 platform. To ensure the robustness of the proposed filters, extensive testing was performed on a diverse set of ECG signal recordings from the MIT-BIH database. The MIT-BIH arrhythmia datasets comprise 48 half-hour excerpts of two-channel ambulatory ECG recordings collected from 47 subjects. The database includes participants of 25 men of age from 32 to 89 years and women ranging from 23 to 89 years of age. The ECG records were digitized at a rate of 360 samples per second, with a resolution of 11 bits over a range of 10 millivolts. Synthetic Power Line Interference (PLI) noise was introduced as a reference signal to test the filtering concept. The proposed filters were rigorously tested and evaluated using many performance metrics, including MSE, SNR, convergence speed, power consumption, and computational complexity in resource utilization of the FPGA platform. These metrics play a crucial role in assessing filtering and feasibility for implementation on an FPGA platform. These specific design metrics were selected due to their interdependence, signifying a tradeoff between them during filter design. The results show that the designed filters successfully reduce the PLI noise and enhance the quality of ECG signals. Furthermore, systolic architecture significantly boosts the filter's performance and reduces power consumption, making them very useful for real-time applications.

The evaluation of the filtering capability of the proposed filters is dependent on the SNR as one of the performance metrics in this study. Table 1 provides the SNR values for ten datasets. The purpose of this analysis is to identify the filters that are most effective in eliminating PLI noise and improving the ECG signal quality. In comparison to LMS without using systolic architecture, the RDLMS filter exhibited a 4.3% improvement in the average SNR. Additionally, the improvement in SNR for the RDRLS filter was 8.2% compared to conventional RLS. Furthermore, the utilization of systolic architecture in the AP filter resulted in a 10% enhancement in SNR compared to the AP filter that does not utilize systolic architecture. These findings demonstrate the notable influence that systolic architecture can have on improving the performance of diverse adaptive filters. The RDLMS filter exhibited a significant improvement in SNR, indicating its potential suitability in situations where reducing noise is of utmost importance. The RDRLS and RDAP filters also demonstrated promising results, highlighting the adaptability of systolic architecture in enhancing the performance of different types of filters. These findings highlight the importance of utilizing systolic architecture during the design and implementation of filters across various applications.

Table 1: A comparative assessment of the Signal-to-Noise Ratio (SNR) in dBs for the entire dataset

Record No.	4-Tap LMS	4-Tap RDLMS	4-Tap RLS	4-Tap RDRLS	4-Tap AP	4-Tap RDAP
data100	31.283	33.392	30.916	33.569	31.593	34.861
data101	31.392	33.173	30.960	33.531	31.690	34.891
data102	30.129	32.692	30.351	33.232	30.230	33.743
data103	26.783	26.419	31.077	33.349	31.809	34.764
data104	30.863	31.953	30.925	33.566	31.615	34.892
data105	31.475	33.456	31.005	33.583	31.797	34.987
data108	31.284	33.391	30.970	33.608	31.572	34.872
data203	27.844	26.976	31.355	33.608	32.574	35.411
data220	31.167	31.633	31.643	33.835	33.405	36.069
data228	30.795	33.048	30.704	33.418	31.005	34.383
Average	30.301	31.613	30.990	33.529	31.729	34.887

Figures 4, 5, and 6 show the convergence rate of the LMS, RLS, and AP filters, both with and without systolic architecture. This comparison provides an important observation regarding systolic architecture's influence on the adaptive filters' performance. The results illustrate an enhancement in the convergence speed. These results indicate that using systolic architecture can also decrease the steady-state Mean Square Error (MSE) of the filter. These findings suggest that the systolic architecture can also reduce the MSE of the adaptive filters.

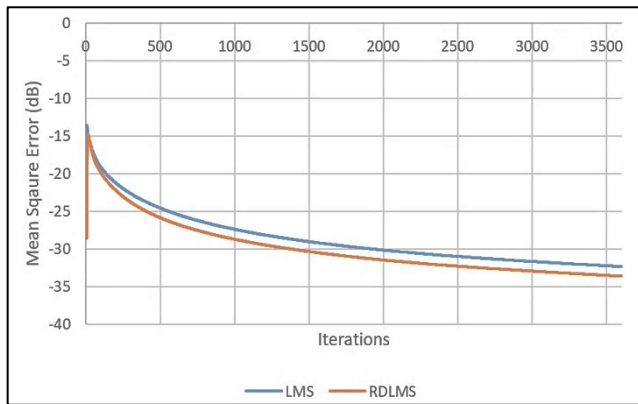
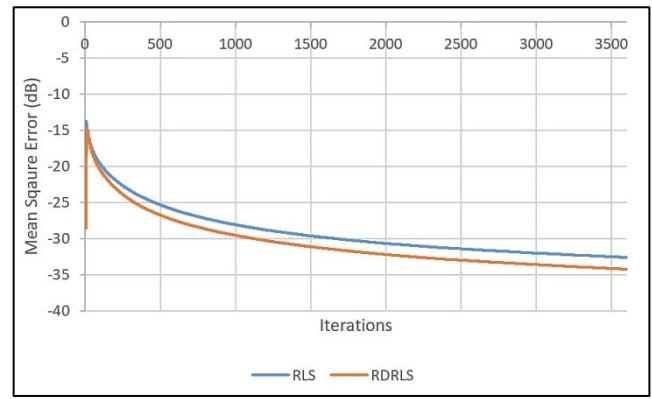
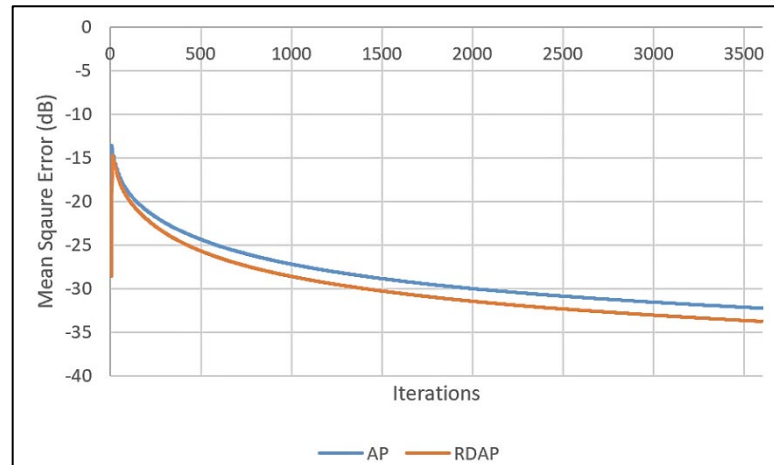

Figure 4: Convergence speed of LMS filter with and without systolic architecture

Figure 5: Convergence speed of RLS filter with and without systolic architecture

Figure 6: Convergence speed of AP filter with and without systolic architecture

Figure 7 showcases the simulation results for removing power line interference (PLI) noise from ECG signal record data 100. Figures 7(a) and 7(b) show clean ECG signal and noisy ECG signal before filtering respectively. The recovered ECG signal underwent RDLMS filtering, resulting in higher accuracy and minimal noise than the recovered ECG signal after LMS filtering, as seen in Figures 7(c) and 7(d). This indicates that RDLMS filtering is a more effective method for enhancing the quality of ECG signals. Furthermore, the recovered ECG signal after RDRLS filtering exhibits high resolution and minimum noise compared to the recovered ECG signal after RLS filtering, as shown in Figures 7(e) and 7(f). This highlights the superiority of RDRLS filtering in enhancing the quality of ECG signals. In addition, RDAP has a higher degree of effectiveness in removing PLI noise than conventional AP filters, as demonstrated in Figures 7(g) and 7(h). This emphasizes the importance of utilizing advanced filtering techniques to improve the accuracy and reliability of ECG signals.

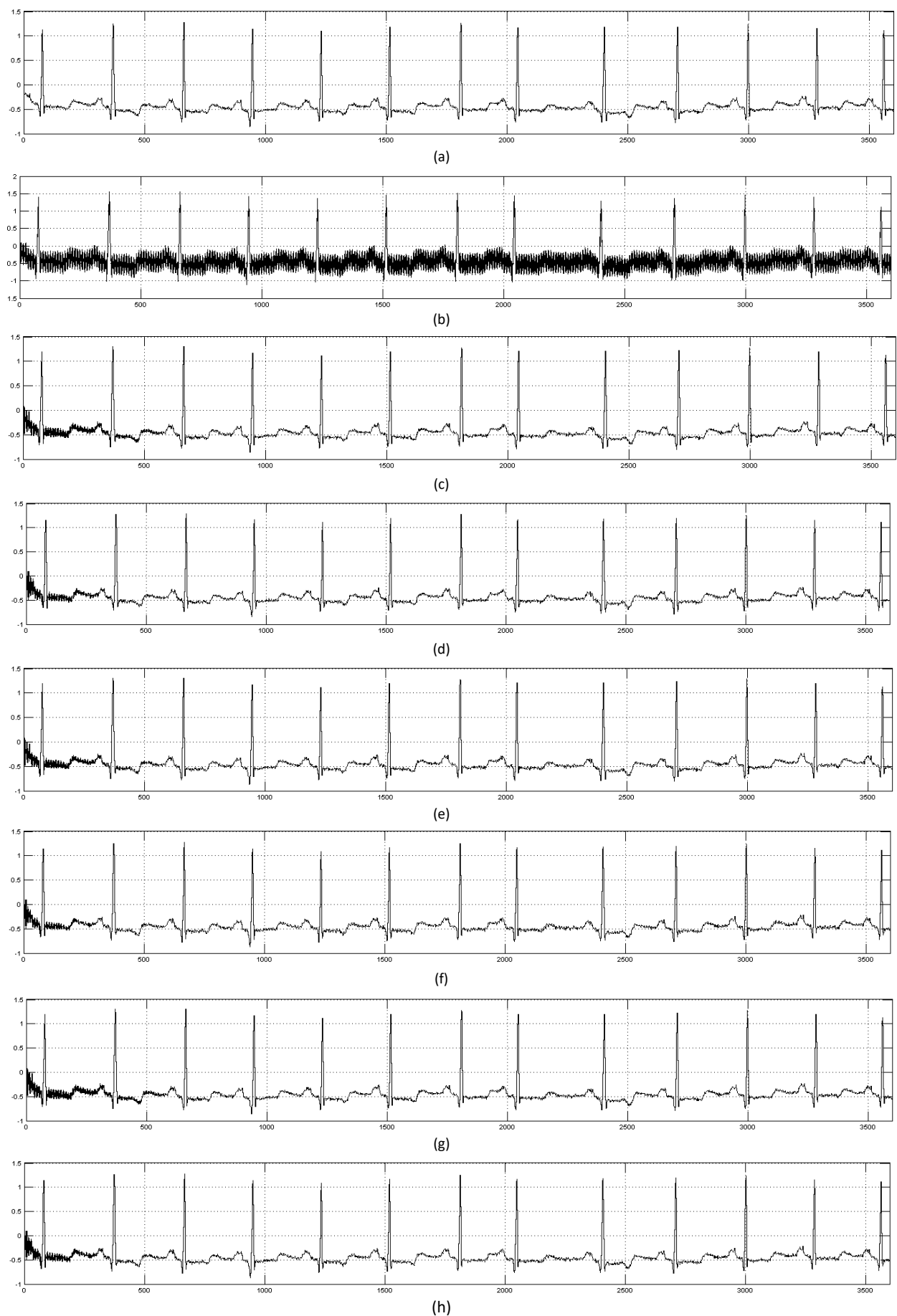


Figure 7: Simulation results for PLI noise removal (a) clean ECG signal, (b) ECG signal with real PLI noise, (c) recovered signal after LMS filtering, (d) recovered signal after RDLMS filtering, (e) recovered signal after RLS filtering, (f) recovered signal after RDRLS filtering, (g) recovered signal after AP filtering, (h) recovered signal after RDAP filtering

Figure 8 illustrates the disparity between ECG clean signals and recovered ECG signals after undergoing LMS, RLS, and AP filtering, both with and without systolic architecture. Upon undergoing RD-based filtering, which utilizes an adaptive filter with systolic architecture, the difference signals exhibit a significantly reduced amount of noise compared to the difference signal after conventional filtering. This suggests that RD-based adaptive filtering is a more efficient method for improving the quality of ECG signals. The RDLMS provide lower residual noise than LMS as shown in Figures 8 (a) and 8 (b). The residual noise after RLS filtering has high amplitude when compare with same filter utilized systolic architecture as in Figures 8 (c) and 8 (d). The remaining noise after RDAP filtering has low level as compare with AP filter as in Figures 8 (e) and 8 (f).

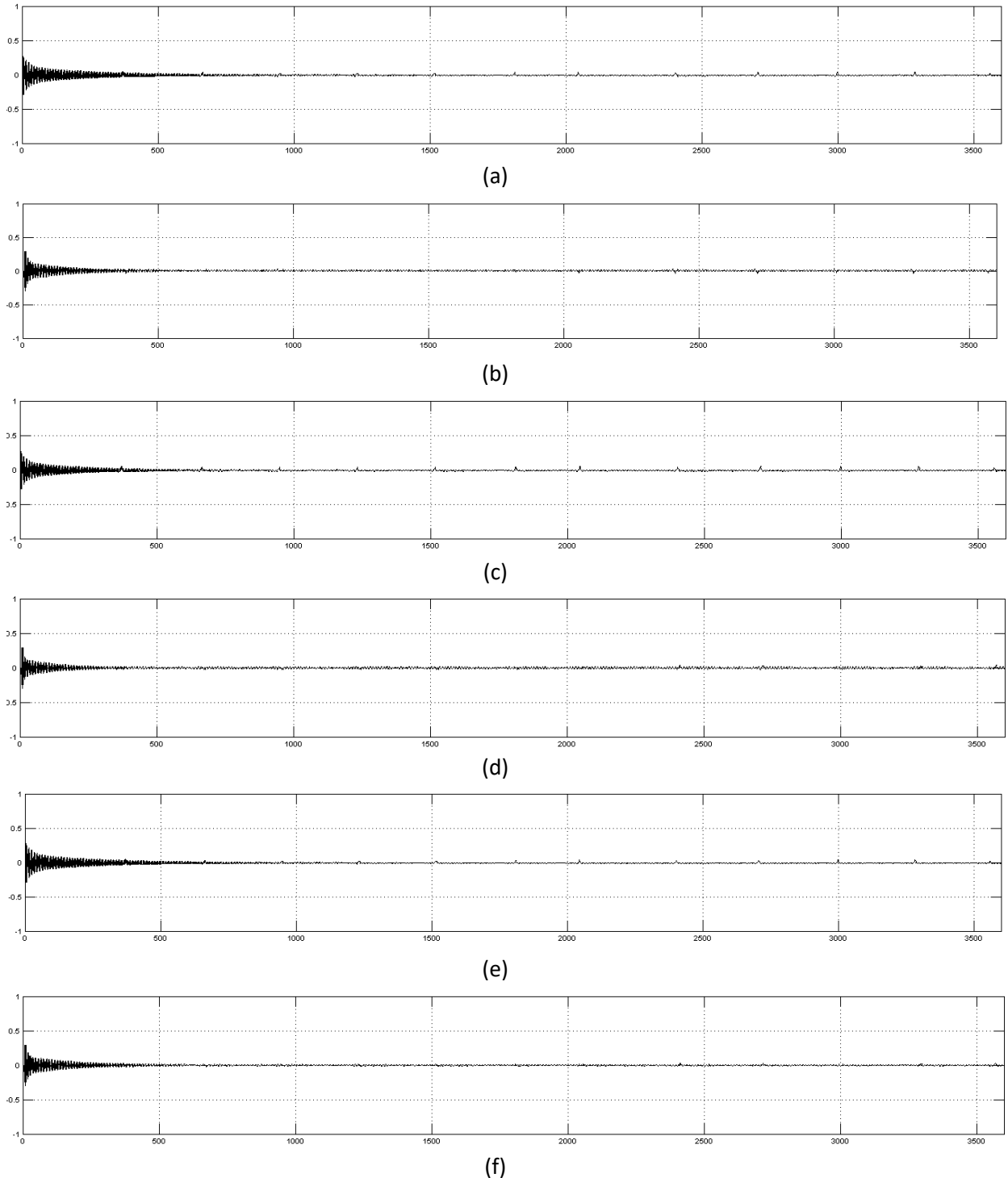


Figure 8: Difference signals for PLI noise removal: (a) Difference signal after performing LMS filtering, (b) Difference signal after performing RDLMS filtering, (c) Difference signal after performing RLS filtering, (d) Difference signal after performing RDRLS filtering, (e) Difference signal after performing AP filtering, (f) Difference signal after performing RDAP filtering

The proposed filter structures were designed using the Xilinx system generator tool in MATLAB 2012a and implemented using the Spartan6 xc6slx16-2csg324 FPGA platform to evaluate their resource utilization. The Xilinx model of the LMS filter with and without systolic architecture is shown in Figures 9 and 10, respectively. The inserted delay block of systolic architecture can be observed by comparing the two figures. The RDLMS filter was designed with many delay blocks to reduce critical path delay to a single multiplier. The resource utilization of both filters is shown in Tables 2 and 3. The LMS design with systolic architecture undoubtedly required more resource area because of the inserted delay. Improving filter performance is more important in real-time signal processing applications.

The percentage of the number of Slice Registers, Slice LUTs, and occupied Slices of RDLMS was increased by 1% from the available resources of the Spartan6 xc6slx16-2csg324 platform. The tables offer valuable insights into the resource requirements of implementing the systolic architecture on the FPGA platform. The systolic architecture enhances filter performance at the cost increases area.

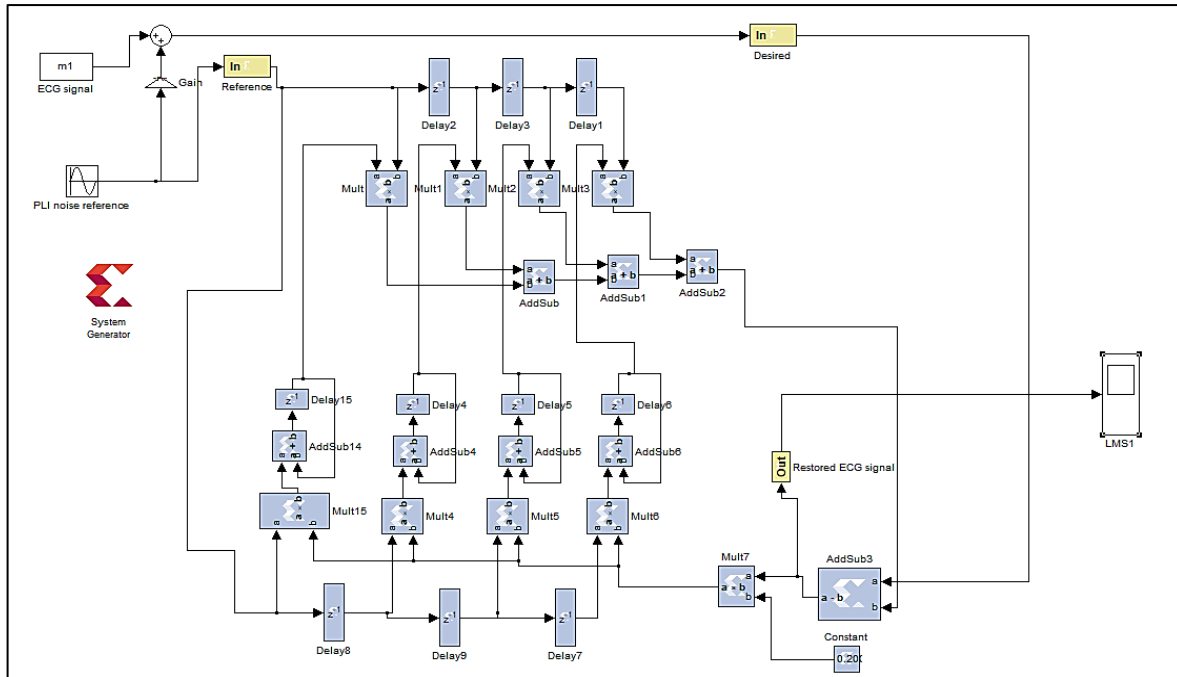


Figure 9: XSG model of 4-Tap LMS filter

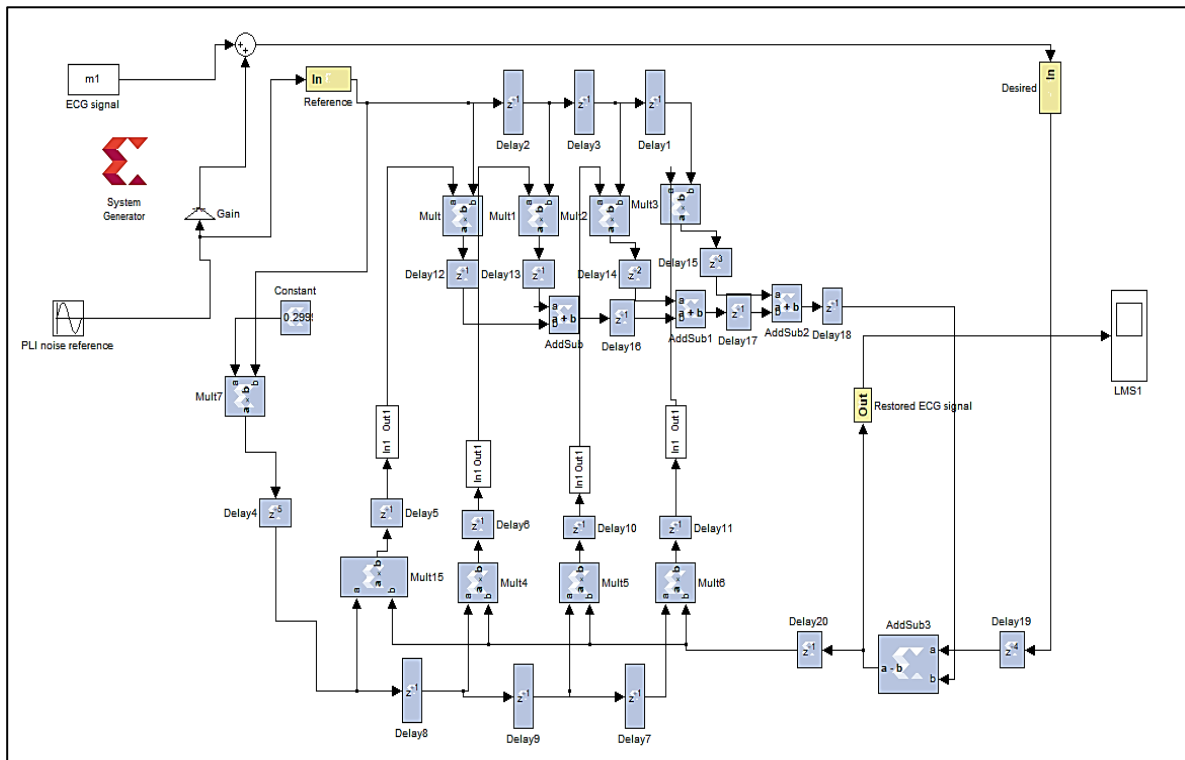


Figure 10: XSG model of 4-Tap RDLMS filter

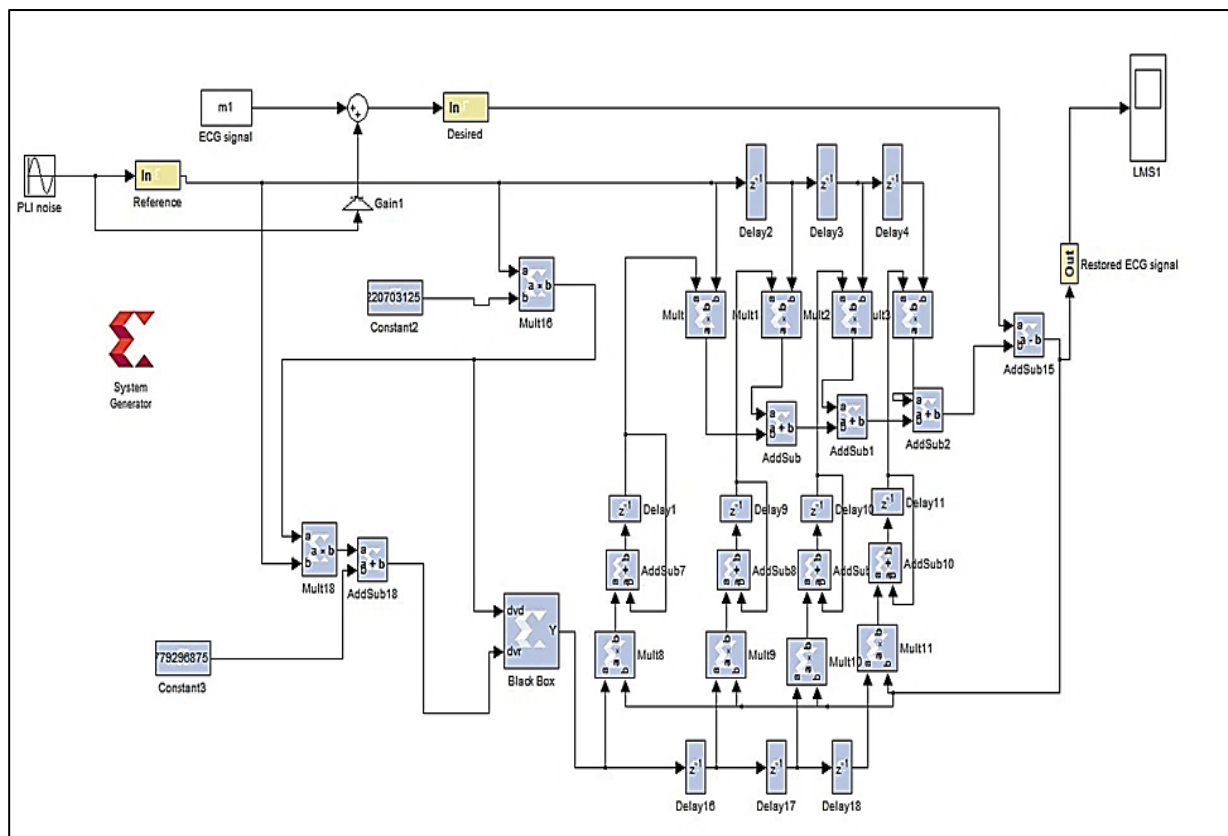
Table 2: Device Utilization Summary of 4-Tap LMS

Slice Logic Utilization	Used	Available	Utilization
Number of Slice Registers	175	18,224	1%
Number of Slice LUTs	173	9,112	1%
Number of occupied Slices	52	2,278	2%
Number of MUXCYs used	132	4,556	2%
Number of LUT Flip Flop pairs used	199		
Number of bonded IOBs	49	232	21%
Number of BUFG/BUFGMUXs	1	16	6%
Number of DSP48A1s	9	32	28%

Table 3: Device Utilization Summary of 4-Tap RDLMS

Slice Logic Utilization	Used	Available	Utilization
Number of Slice Registers	384	18,224	2%
Number of Slice LUTs	225	9,112	2%
Number of occupied Slices	73	2,278	3%
Number of MUXCYs used	132	4,556	2%
Number of LUT Flip Flop pairs used	287		
Number of bonded IOBs	49	232	21%
Number of BUFG/BUFGMUXs	1	16	6%
Number of DSP48A1s	9	32	28%

The 4-Tap RLS filter was also designed using the XSG tool, as shown in Figure 11. This filter contains division terms in the weights update equation. The non-restoring algorithm has been used to perform the division task in the FPGA platform. Figure 12 shows the XSG model of the 4-Tap RLS filter designed with systolic architecture. The RDLMS filter required the same number of inserted delays as the LMS filter because both filters were designed with direct form FIR filters, as shown in Figure 12, but the RLS filter required more area than the LMS filter because RLS is more complex. The resource utilization of RLS and RDLMS filters was demonstrated in Tables 4 and 5, respectively.


Figure 11: XSG model of 4-Tap RLS filter

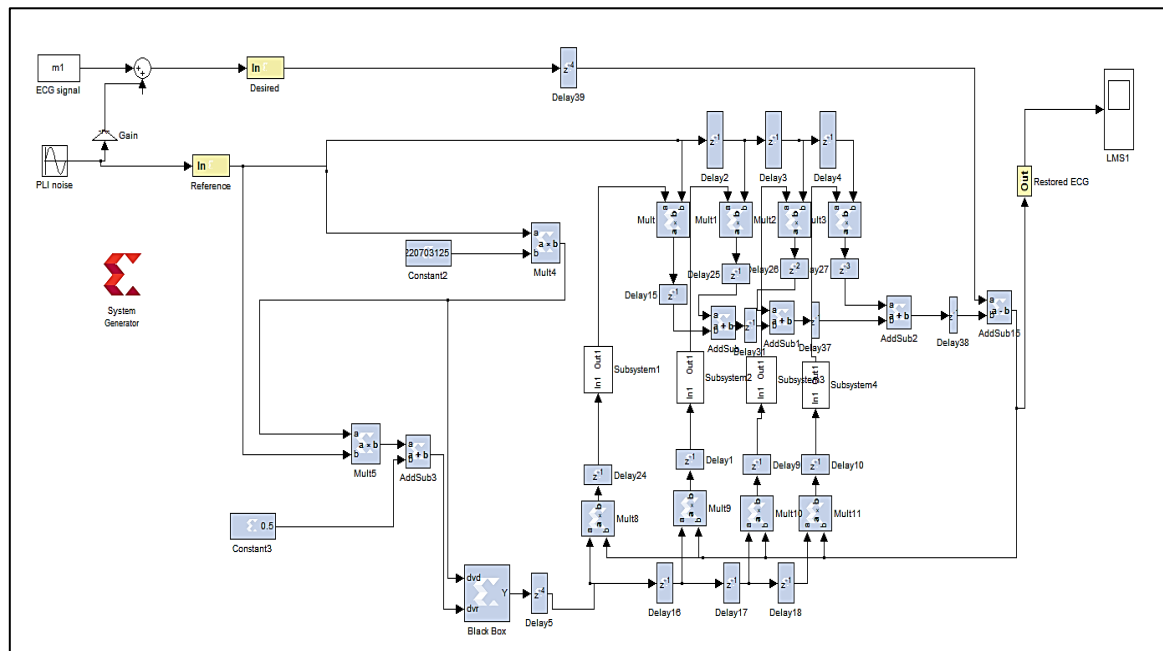


Figure 12: XSG model of 4-Tap RDRLS filter

Table 4: Device Utilization Summary of 4-Tap RLS

Slice Logic Utilization	Used	Available	Utilization
Number of Slice Registers	208	18,224	1%
Number of Slice LUTs	1,428	9,112	15%
Number of occupied Slices	478	2,278	20%
Number of MUXCYs used	1,172	4,556	25%
Number of LUT Flip Flop pairs used	1,495		
Number of bonded IOBs	49	232	21%
Number of BUFG/BUFGMUXs	1	16	6%
Number of DSP48A1s	14	32	43%

Table 5: Device Utilization Summary of 4-Tap RDRLS

Slice Logic Utilization	Used	Available	Utilization
Number of Slice Registers	432	18,224	2%
Number of Slice LUTs	1,518	9,112	16%
Number of occupied Slices	502	2,278	22%
Number of MUXCYs used	1,168	4,556	25%
Number of LUT Flip Flop pairs used	1,590		
Number of bonded IOBs	49	232	21%
Number of BUFG/BUFGMUXs	1	16	6%
Number of DSP48A1s	14	32	43%

The percentage of the number of Slice Registers and Slice LUTs RDRLS was increased by 1%, while the number of occupied Slices was increased by 2% from the available resources of the Spartan6 xc6slx16-2csg324 platform.

4-Tap AP adaptive filter has an inverse term in the weights update equation, which is performed in this work using a non-restoring divider with constant dividend, the XSG modal of AP filter shown in Figure 13. This filter performs efficiently in terms of SNR, as shown in the previous chapter, while the hardware resources and power consumption are discussed in this chapter. The retimed delay-based AP filter XSG model is shown in Figure 14. Compared with LMS and RLS filters, the AP filter required the same number of inserted delay blocks when applying systolic architectures. The resource utilization of both AP and RDAP filters are shown in Tables 6 and 7, respectively.

The percentage of the number of Slice Registers RDRLS was increased by 1%. In comparison, the number of Slice LUTs and the number of occupied Slices were increased by approximately less than 1% from the available resources of the Spartan 6 xc6slx16-2csg324 platform.

Table 8 demonstrates the power consumption of various filter structures. Employing systolic architecture in the LMS filter reduces power consumption by 40.5% compared with conventional LMS filters. Utilizing systolic architecture in an RLS filter reduces power consumption by about 28.6%. Utilizing systolic architecture in an AP filter reduces power consumption by about 5.9%. These results demonstrate that the systolic architecture significantly reduces power consumption in adaptive filters. It is noteworthy that the implementation of systolic architecture significantly impacts the power consumption of adaptive filters.

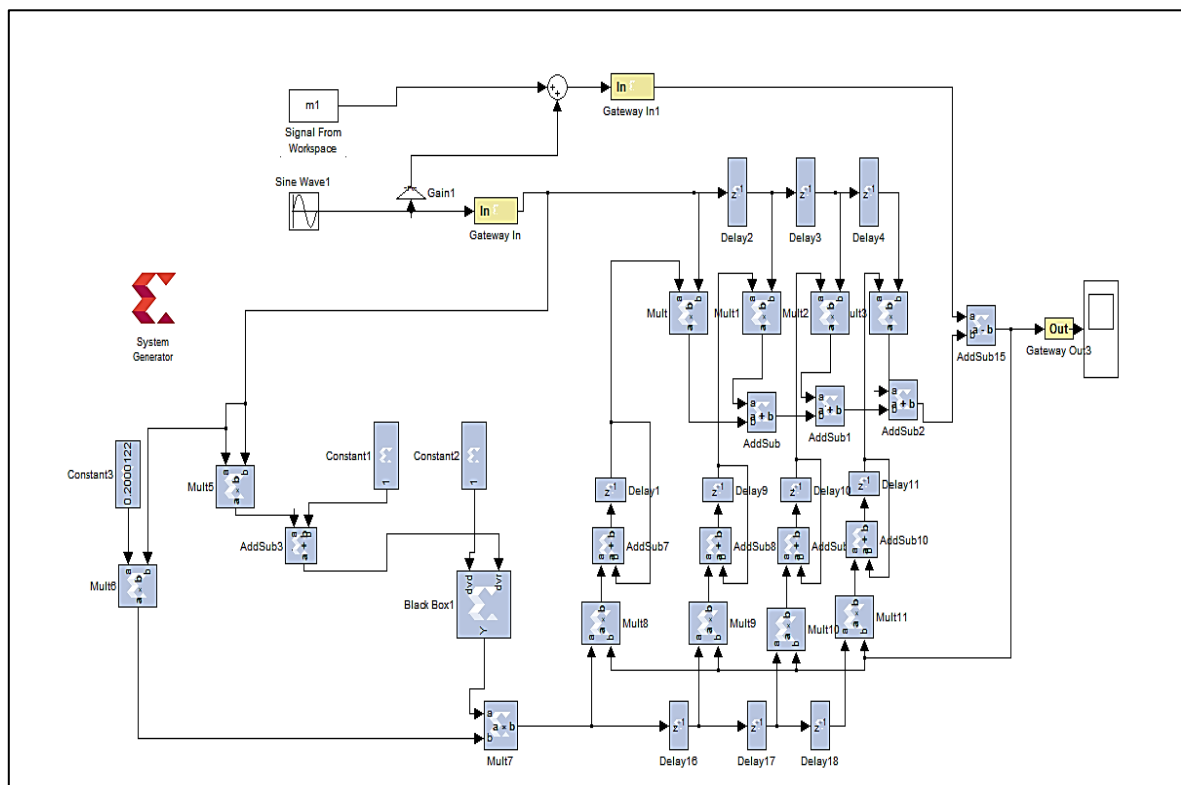


Figure 13: XSG model of 4-Tap AP filter

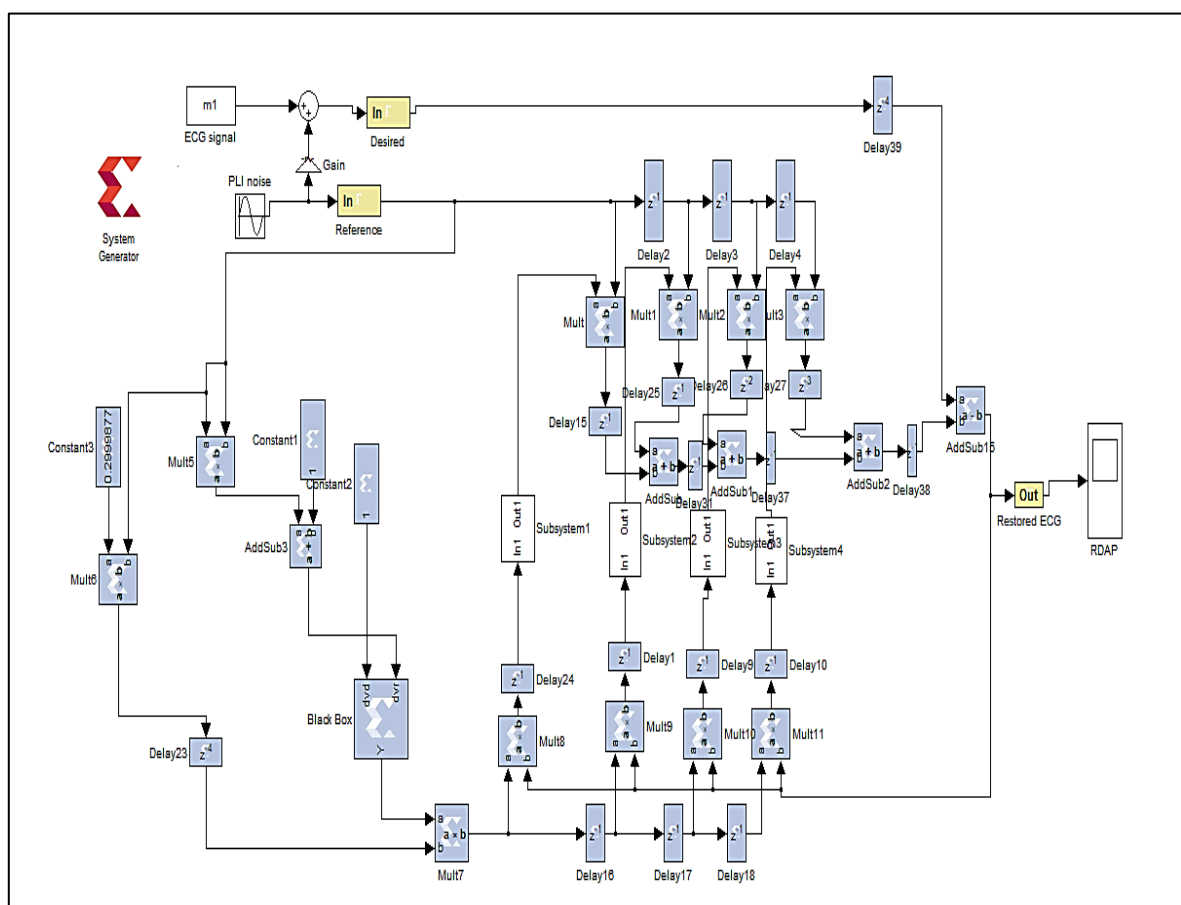


Figure 14: XSG model of 4-Tap RDAP filter.

Table 6: Device Utilization Summary of 4-Tap AP

Slice Logic Utilization	Used	Available	Utilization
Number of Slice Registers	160	18,224	1%
Number of Slice LUTs	1,388	9,112	15%
Number of occupied Slices	466	2,278	20%
Number of MUXCYs used	1,144	4,556	25%
Number of LUT Flip Flop pairs used	1,419		
Number of bonded IOBs	49	232	21%
Number of BUFG/BUFGMUXs	1	16	6%
Number of DSP48A1s	12	32	37%

Table 7: Device Utilization Summary of 4-Tap RDAP

Slice Logic Utilization	Used	Available	Utilization
Number of Slice Registers	368	18,224	2%
Number of Slice LUTs	1,427	9,112	15%
Number of occupied Slices	469	2,278	20%
Number of MUXCYs used	1,140	4,556	25%
Number of LUT Flip Flop pairs used	1,516		
Number of bonded IOBs	49	232	21%
Number of BUFG/BUFGMUXs	1	16	6%
Number of DSP48A1s	12	32	37%

Table 8: Power consumption comparison of various filter structures with and without systolic Architecture.

Filter type	4-Tap LMS	4-Tap RDLMS	4-Tap RLS	4-Tap RDRLS	4-Tap AP	4-Tap RDAP
Power in watt	0.2	0.119	0.258	0.184	0.184	0.173

7. Conclusion

The proposed filters make it possible to reduce the tradeoff between filter characteristics, which is very important in real-time applications, especially ECG signal processing. The filter characteristic tradeoff is the main design problem in adaptive filter applications. In any attempt to increase the convergence speed by varying filter parameters, the SNR reduces and results in bad filtering capability. The outcome of using systolic architecture in filter design increases convergence speed and SNR. Moreover, this paper has proven systolic architecture to be an efficient architecture for designing energy-efficient adaptive filters. The power consumption of all filters designed in this paper was reduced by a significant level. The reduction in power consumption observed in RDLMS, RDRLS, and RDAP is a testament to the effectiveness of this architecture. The improvement in SNR was 4.3%, 8.2%, and 10% for RDLMS, RDRLS, and RDAP filters, respectively. The power consumption was reduced by about 40.5%, 28.6%, and 5.9% for RDLMS, RDRLS, and RDAP filters. Moreover, the results show significant improvement in the convergence speed of the proposed filters when compared with the same filter designed without systolic architecture. The results demonstrate that despite increasing resource utilization, the proposed filter designed with systolic architecture is more efficient for denoising ECG signals from PLI noise. This is because improving filter performance is of greater importance for DSP applications. However, the utilization of resources indicates that the proposed filters are implementable in the FPGA platform. By utilizing the non-restoring division algorithm, the design can achieve a highly efficient and accurate implementation of the division term.

Author contributions

Conceptualization, H. Thannoon and I. Hashim; methodology, H. Thannoon and I. Hashim; software, H. Thannoon and I. Hashim; validation, H. Thannoon and I. Hashim; formal analysis, H. Thannoon and I. Hashim; investigation, H. Thannoon and I. Hashim; resources, H. Thannoon and I. Hashim; writing—original draft preparation, H. Thannoon and I. Hashim ; writing—review and editing, H. Thannoon and I. Hashim; visualization, H. Thannoon and I. Hashim; supervision, I. Hashim; project administration, H. Thannoon and I. Hashim. All authors have read and agreed to the final version of the manuscript.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Data availability statement

The data supporting this study's findings are available on request from the corresponding author.

Conflicts of interest

The authors declare that there is no conflict of interest.

References

- [1] D. Jain, P. Beniwal, Review Paper on Noise Cancellation using Adaptive Filters, *Int. J. Eng. Res. Technol.*, 11 (2022) 241-244.
- [2] A. Dickman, Adaptive Filters. In: *Verified Signal Processing Algorithms in MATLAB and C*, Springer, BOOK, 2022.
- [3] X. Wang, A. Wang, D. Wang, W. Wang, An improved spline adaptive filter for nonlinear system identification under impulsive noise environment, *Energy Rep.*, 8 (2022) 832-840. <https://doi.org/10.1016/j.egyr.2021.11.068>
- [4] A. Z. Khan, I. Shafi, Removing Artifacts from Raw Electrocardiogram Signals Using Adaptive Filter in State Space. *Circuits Syst. Signal Process.*, 39 (2020). <https://doi.org/10.1007/s00034-019-01149-3>
- [5] T. M. Jamel, K. K. Al-Magazachi, Simple variable step size LMS algorithm for adaptive identification of IIR filtering system, *The 5th International Conference on Communications, Computers and Applications (MIC-CCA2012)*, 2012, 23-28.
- [6] M. Kazemi, M. M. Arefi, A fast iterative recursive least squares algorithm for Wiener model identification of highly nonlinear systems, *ISA Transactions*, 67 (2017) 382-388. <https://doi.org/10.1016/j.isatra.2016.12.002>
- [7] A.O. Abid Noor, Adaptive Noise Cancellation Using Noise Dependent Affine Projection Algorithm, *Eng. Technol. J.*, 35 (2017) 660-667. <https://doi.org/10.30684/etj.35.6A.15>
- [8] S. Gao, H. Yoshida, K. Seto, S. Komatsu, M. Fujita, Interconnect-aware pipeline synthesis for array based architectures, *Trans. Fundam. Electron. Com-mun. Comp. Sci.*, 92 (2009) 1464–1475. <http://dx.doi.org/10.1587/transfun.E92.A.1464>
- [9] M. Bahoura, H. Ezzaidi, FPGA-Implementation of Parallel and Sequential Architectures for Adaptive Noise Cancellation, *Circuits Syst. Signal Process.*, 30 (2011) 1521-1548. <https://doi.org/10.1007/s00034-011-9310-0>
- [10] G. Swaminathan, G. Murugesan, S. Sasikala, L. Murali, A novel implementation of combined systolic and folded architectures for adaptive filters in FPGA, *Microprocess. Microsyst.*, 74 (2020). <https://doi.org/10.1016/j.micpro.2020.103018>
- [11] M. Jayapraivintha, G. Swaminathan, G. Murugesan, Design of Systolic architecture for various adaptive filters for noise cancellation, 2015 3rd International Conference on Signal Processing, Commun. Netw., Chennai, India, (2015) 1-6. <https://doi.org/10.1109/ICSCN.2015.7219907>
- [12] M. Chandra, P. Goel, A. Anand, A. Kar, Design and analysis of improved high-speed adaptive filter architectures for ECG signal denoising, *Biomed. Signal Process. Control.*, 63 (2021). <http://dx.doi.org/10.1016/j.bspc.2020.102221>
- [13] V. Kavitha, P. Kaviya Priya, Tha. Sugapriya, Efficient Implementation of Adaptive Filter Architecture Using Gate Level Modification for ECG Denoising, *Proceedings of 2018 the 8th International Workshop on Computer Science and Engineering*, Bangkok, 28 (2028) 171-177. <http://dx.doi.org/10.18178/wcse.2018.06.031>
- [14] A. A. M. Khalaf, M. M. Ibrahim, H. F. A. Hamed, Performance study of adaptive filtering and noise cancellation of artifacts in ECG signals, 2015 17th International Conference on Advanced Communication Technology (ICACT), Pyeong Chang, Korea (South), 2015, 394-401. <http://dx.doi.org/10.1109/ICACT.2015.7224826>
- [15] S. Veni, Real Time Implementation of SIGN LMS Adaptive Filters using Xilinx System Generator, *Int. J. Math. Comput. Simul.*, 14 (2020) 8-15.
- [16] J. A. Van Alste, T. S. Schilder, Removal of Base-Line Wander and Power-Line Interference from the ECG by an Efficient FIR Filter with a Reduced Number of Taps, *IEEE Trans. Biomed. Eng.*, BME-32 (1985) 1052-1060. <http://dx.doi.org/10.1109/TBME.1985.325514>
- [17] T. S. Ghmati, A. A. S. Elhoula, Adaptive Digital FIR Filters: Case Study: Noise Cancellation using LMS Algorithm, *Albahit J. Appl. Sci.*, 2 (2021) 30-36
- [18] H. F. Yahya, I. A. Hashim, Enhanced Shifting Method For an Area-Efficient Design of FIR Filter Based on FPGA, *5th Int. Conf. Eng. Technol. Appl.*, (2022) 414-418. <http://dx.doi.org/10.1109/ICETA54559.2022.9888340>
- [19] F. Ding, Y. Wang, J. Ding, Recursive least squares parameter identification algorithms for systems with colored noise using the filtering technique and the auxiliary model, *Digital Signal Process.*, 37 (2015) 100-108. <https://doi.org/10.1016/j.dsp.2014.10.005>
- [20] Y. Hu, Iterative and recursive least squares estimation algorithms for moving average systems, *Simul. Modell. Pract. Theory*, 34 (2013) 12-19. <https://doi.org/10.1016/j.simpat.2012.12.009>
- [21] M. V. Sudhakar, M. P. Charan, G. N. Pranai, L. Harika, P. Yamini, Audio signal noise cancellation with adaptive filter techniques, *Materials Today: Proceedings*, 80 (2023) 2965-2963. <https://doi.org/10.1016/j.matpr.2021.07.080>
- [22] Z. Zhu, X. Gao, L. Cao, D. Pan, Y. Cai, Y. Zhu, Analysis on the adaptive filter based on LMS algorithm, *Optik*, 127 (2016) 4698-4704. <https://doi.org/10.1016/j.ijleo.2016.02.005>

- [23] Th. M. Jamel, H. A. Mohamed, Noise Canceller Using a New Modified Adaptive Step Size LMS Algorithm, *Wseas Transactions On Signal Processing*, 10 (2014) 637-644.
- [24] D. Liu, H. Zhao, Affine Projection Sign Subband Adaptive Filter Algorithm With Unbiased Estimation Under System Identification, *IEEE Trans. Circuits Syst. II Express Briefs*, 70 (2023) 1209-1213. <https://doi.org/10.1109/TCSII.2022.3216807>
- [25] Y. Ren, Y. Zhi, J. Zhang, Geometric-algebra affine projection adaptive filter, *EURASIP J. Adv. Signal Process*, 82 (2021). <https://doi.org/10.1186/s13634-021-00790-y>
- [26] S. Jayapoorani, D. Pandey, N. S. Sasirekha, et al, Systolic optimized adaptive filter architecture designs for ECG noise cancellation by Vertex-5. *AS 6* (2023) 163–173. <https://doi.org/10.1007/s42401-022-00177-3>
- [27] C. P. Chowdari, J. B. Seventline, Systolic architecture for adaptive block FIR filter for throughput using distributed arithmetic, *Int. J. Speech Technol.*, 23 (2020) 549–557. <https://doi.org/10.1007/s10772-020-09745-4>