www.czasopisma.pan.pl PAN POLSKA AKADEMIA NAUK www.journals.pan.pl

jet INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2023, VOL. 69, NO. 4, PP. 811-817
Manuscript received June 8, 2023; revised October, 2023. DOI: 10.24425/ijet.2023.147705

Efficient FPGA Implementation of Recursive Least Square Adaptive Filter Using Non-Restoring Division Algorithm

Harith H. Thannoon, and Ivan A. Hashim

***Abstract*—In this paper, Recursive Least Square (RLS) and Affine Projection (AP) adaptive filters are designed using Xilinx System Generator and implemented on the Spartan6 xc6slx16-2csg324 FPGA platform. FPGA platform utilizes the non-restoring division algorithm and the COordinate Rotation DIgital Computer (CORDIC) division algorithm to perform the division task of the RLS and AP adaptive filters. The Non-restoring division algorithm demonstrates efficient performance in terms of convergence speed and signal-to-noise ratio. In contrast, the CORDIC division algorithm requires 31 cycles for division initialization, whereas the non-restoring algorithm initializes division in just one cycle. To validate the effectiveness of the proposed filters, a set of ten ECG records from the BIT-MIT database is used to test their ability to remove Power Line Interference (PLI) noise from the ECG signal. The proposed adaptive filters are compared with various adaptive algorithms in terms of Signal-to-Noise Ratio (SNR), convergence speed, residual noise, steady-state Mean Square Error (MSE), and complexity.**

***Keywords*—Adaptive filter; RLS; AP; CORDIC; non-restoring**

I. Introduction

An adaptive filter is a crucial tool in the heart disease diagnosis system. To ensure the efficient and accurate operation of the diagnosis system, the ECG signal must be clean and have high resolution [1]. In practical scenarios, the ECG signal is often interfered with by various types of noise signals. Conventional filters, including non-adaptive filters, have been used to restore the ECG signal but have also destroyed some required information, leading to misinterpretation of certain heart diseases. The adaptive filter is the most common approach for efficiently denoising the ECG signal without distorting the required information. It is used in conjunction with diagnosis systems to achieve accurate results [2].

Several adaptive algorithms have been used in adaptive filters for noise cancellers, including the Least Mean Square (LMS) [3], Recursive Least Square (RLS) [4], and Affine Projection (AP) algorithms [5]. The LMS-based adaptive filter has a simple construction but suffers from a low convergence speed. On the other hand, RLS and AP algorithms have a high convergence speed with more complexity. However, the RLS algorithm faces difficulties in design and implementation on the FPGA platform because it has two update equations, one of which has a nonzero initial value, while XSG always initializes variables and parameters from zero. Moreover, the RLS and AP algorithms have a division term in their weights update equations and require an efficient division algorithm to perform this task. These issues related to RLS and AP are the main focus of this paper.

Many researchers have designed RLS and AP algorithms as noise cancellers to achieve high convergence speed. Here are some related works to this paper: Gomathi Swaminathan et al. [6] designed an RLS adaptive filter for ECG noise cancellation using XSG. However, they performed the division terms outside the XSG tool by using a divider from MATLAB Simulink. Additionally, their proposed filter considered the cross-correlation matrix of the RLS algorithm as constant, which led to poor performance. Jayapravintha M. et al. [7] designed RLS and AP adaptive filters using the XSG tool but faced similar problems as mentioned earlier for the RLS algorithm. Moreover, they designed the division term of the AP algorithm outside the XSG tool, which does not provide a real implementation of the AP algorithm in the FPGA platform, nor the actual resource utilization of the AP algorithm. V. Kavitha et al. [8] also designed an RLS adaptive filter with a constant cross-correlation matrix and the division term performed outside the XSG tool. This design does not consider an efficient RLS adaptive filter and does not provide the actual utilization of the FPGA platform.

Authors in [9], [10],[11],[12] designed adaptive filters for ECG noise cancellation, but none of them mentioned the aforementioned problems. Most of them used the LMS algorithm because of its simple structure and ease of design using the XSG tool. However, the LMS algorithm suffers from low convergence speed. Furthermore, the adaptive filters designed in the previous works were not validated based on convergence speed, which is an important metric for testing the filter's performance in noise removal.

To address the aforementioned problems related to RLS and AP algorithms, both filters were designed in the XSG tool. The division term of the filters was implemented using two different division algorithms: CORDIC [13] and non-restoring [14] algorithms, specifically tailored for the FPGA platform. The performance of the AP and RLS adaptive filters was compared using these two division algorithms in terms of convergence speed, SNR, signal resolution, and the amount of residual noise in the ECG signal. Furthermore, the RLS algorithm was enhanced by incorporating an updated cross-correlation matrix to further improve the performance of the adaptive filter.

Harith H. Thannoon and Ivan A. Hashim are with the University of Technology, Iraq (e-mail:eee.20.01@grad.uotechnology.edu.iq, ivan.a.hashim @uotechnology.edu.iq).

II. Adaptive Filter Algorithms

A. *Recursive Least Square algorithm.*

The Recursive Least Squares (RLS) algorithm is an adaptive algorithm used to update filter coefficients recursively based on minimizing the Mean Square Error (MSE) at the output of the noise canceller. This algorithm demonstrates a faster convergence speed, each input sample was processed in a recursive manner at each iteration. The RLS performs efficiently in non-stationary environments where the noise signal varies with time. A set of equations below present the principle of this algorithm [15].

$$e(n) = d(n) - y(n) = d(n) - x^T(n) * w(n) \quad (1)$$
$$w(n+1) = w(n) + e(n) * g(n) \quad (2)$$
$$g(n) = P(n) * x(n)\{\lambda + x^T(n) * P(n) * x(n)\}^{-1} \quad (3)$$
$$P(n+1) = \lambda^{-1} * P(n) - g(n) * x^T(n) * \lambda^{-1} * P(n) \quad (4)$$

In these equations, y(n) is the filter output that represents convolution between the input signal x(n), and the weights vector denoted by w(n). The error signal e(n) determined by the difference between the desired signal d(n) and the output signal. λ denoted forgetting factor the govern the filter performance, this parameter affects both signal-to-noise ratio (SNR) and convergence speed. 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 [16].

B. *Affine Projection algorithm.*

The AP algorithm plays a crucial role in improving the efficiency of corrupted signals. it estimates the coefficients of the filter using a set of desired and input signals. The main advantage of this algorithm is its capability to eliminate time-varying noises, which are typically challenging to eliminate through conventional filters. This advantage makes it a suitable choice for applications including signal enhancement, where the input signal can vary widely in both frequency and amplitude. The following fundamental equations constitute the AP filter. [17].

$$y(n) = d(n) - x^T(n) * w(n) \quad (8)$$
$$e(n) = d(n) - y(n) \quad (9)$$
$$w(n+1) = w(n) + \big(\mu * x(n)\big)[x^T(n) * x(n) + \delta I]^{-1} * e(n) \quad (10)$$

In this context, e(n) represents the output error of the noise canceller, while d(n) denotes the desired signal, and w(n) represents the weight vector of the filter. The input signal is represented by x(n), and the step size parameter is denoted by μ, while the regularization parameter is represented by δ. The regularization parameter δ and the step size parameter μ are crucial in this process as they influence the performance of the filter. The value of δ helps to prevent overfitting and improve the model's generalization, while μ determines the rate at which the filter reaches its steady state, significantly impacting the learning process's speed and accuracy [18].

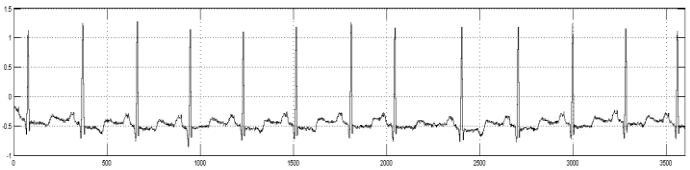
III. Results and Discussion

In this paper, 4-tap AP and RLS adaptive filters are designed using the XSG tool and implemented on the Spartan6 xc6slx16-2csg324 FPGA platform. The proposed adaptive filter was tested and validated to remove PLI noise from ECG signals. The synthetic PLI noise was obtained from a signal generator that generates a sinusoidal signal with a frequency of 60 Hz. The ECG signals were obtained from MIT-BIH datasets, which include several ECG signals with a wide variety of signal morphologies.

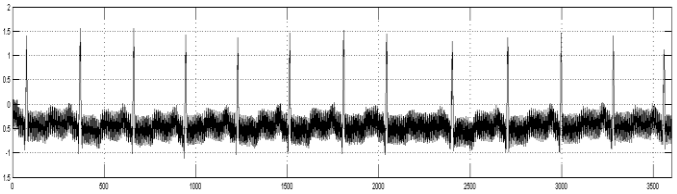
Fig. 1 depicts the ECG signals that have been restored using different algorithms. This figure provides valuable insight into the effectiveness of the designed filters in maintaining the quality of the ECG signal. Fig. 2 shows the difference between the recovered signal and the clean ECG signal after applying different algorithms. The adaptive algorithms modify the filter coefficients based on the input signal to minimize the difference between the desired and actual output. Measuring the signal difference provides insight into the algorithm's effectiveness in removing unwanted components or noise from the original signal. A smaller difference signal indicates better noise reduction and more precise signal processing. The ECG signal quality after non-restoring-based RLS and AP adaptive filters has better resolution when compared with the restored signal after the CORDIC-based adaptive filter, as shown in Figs. 1 (g), 1 (h), 1 (j), and 1 (k). Moreover, the difference signals after non-restoring-based RLS and AP adaptive filters contain less residual noise than the difference signals after CORDIC-based filters, as shown in Figs. 2 (e), 2 (f), 2 (h), and 2 (i).

The main advantage of using non-restoring division over CORDIC division algorithms is that it does not require a long time to initialize the division process, as in CORDIC, which requires approximately 31 delay cycles to initiate the division algorithm. During the first 31 cycles, the CORDIC algorithm provides wrong division results, which affect the final output of the filters and introduce a large error in the first samples. The RLS filter incorporating an updated cross-correlation matrix and designed using the MATLAB divider outside the XSG tool provides the minimum residual noise, as shown in Figs. 1 (i) and 2 (g). The non-restoring division algorithm has efficiency close to the MATLAB divider, as shown in Figs. 1 and 2, making it the best choice for performing the division task in FPGA-based adaptive filters.

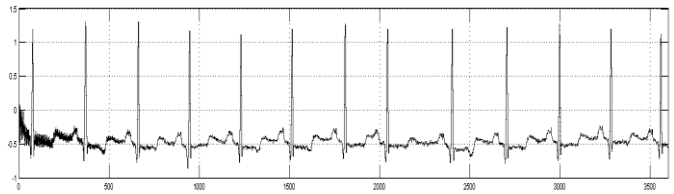
The XSG models of the proposed filters are shown in Figs. 3, 4, and 5. Fig. 3 shows the XSG model of the 4-tap AP filter design using a non-restoring divider, Fig. 4 shows the XSG model of the RLS filter incorporating a constant cross-correlation matrix with a non-restoring divider, and Fig. 5 shows the XSG model of the proposed 4-tap RLS filter incorporating an updated cross-correlation matrix with a non-restoring divider.



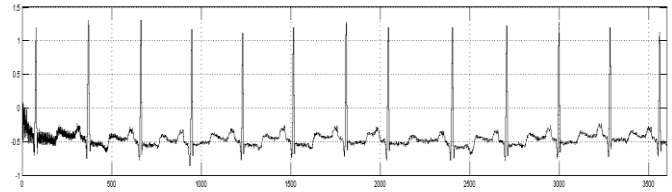
(a)



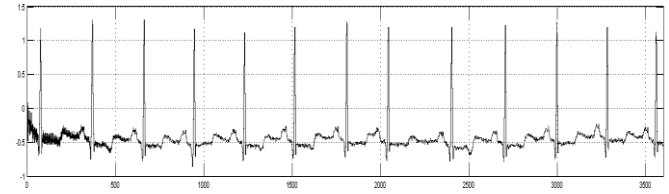
(b)



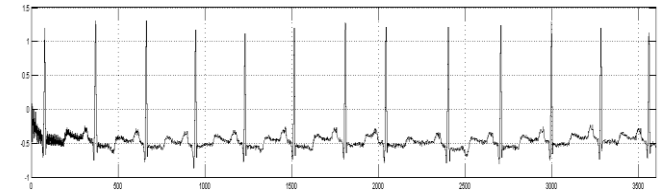
(c)



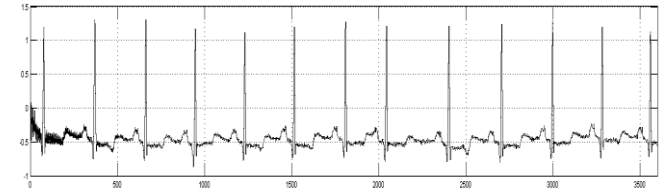
(d)



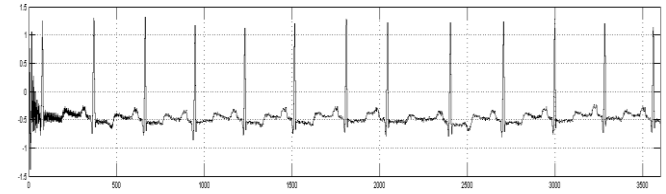
(e)



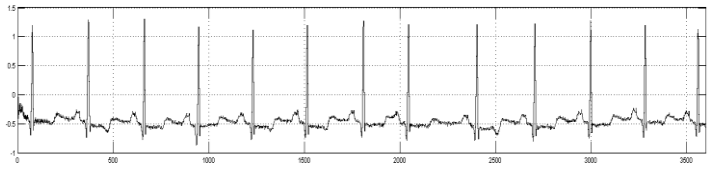
(f)



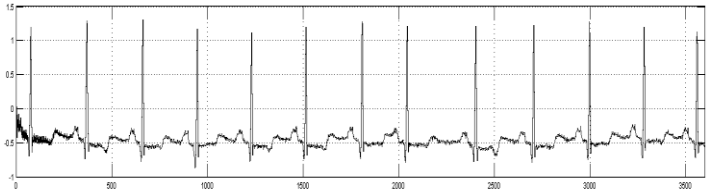
(g)



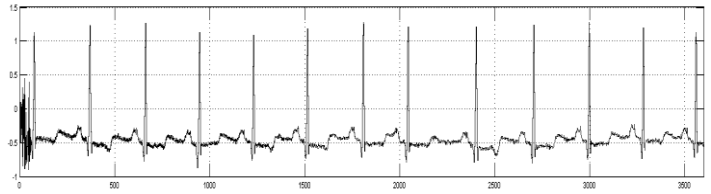
(h)



(i)

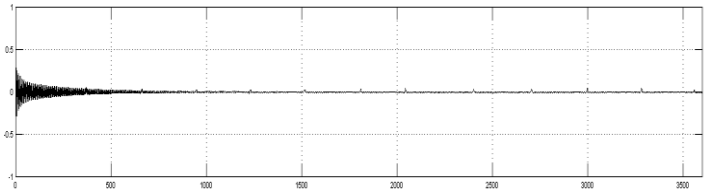


(j)

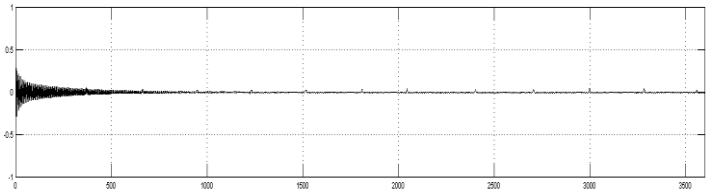


(k)

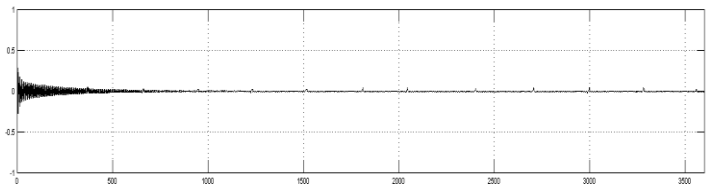
Fig.1. Simulation results for PLI noise removal(a) clean ECG signal, (b) ECG signal with real PLI noise, (c) recovered signal after AP filtering, (d) recovered signal after AP design with non-restoring division algorithm, (e) recovered signal after AP design with CORDIC division algorithm, (f) recovered signal after RLS with constant cross correlation matrix, (g) recovered signal after RLS design with constant cross correlation matrix and non-restoring division algorithm, (h) recovered signal after RLS design with constant cross correlation matrix and CORDIC division algorithm, (i) recovered signal after RLS filtering designed with update cross-correlation matrix, (j) recovered signal after RLS filtering designed with update cross-correlation matrix and non-restoring division algorithm, (k) recovered signal after RLS filtering designed with update cross-correlation matrix and CORDIC division algorithm.



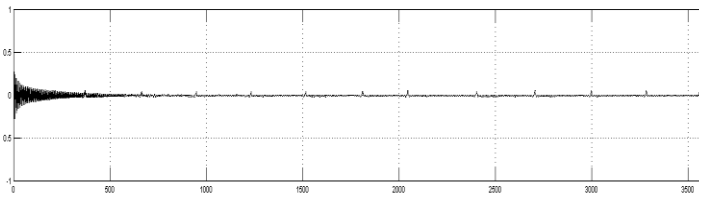
(a)



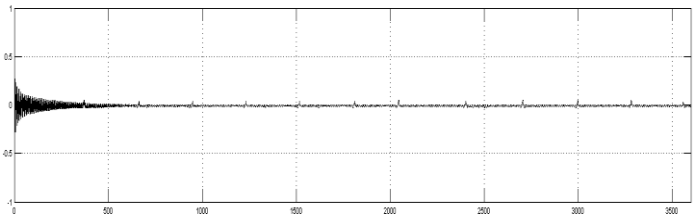
(b)



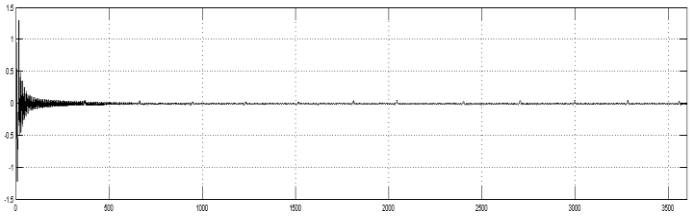
(c)



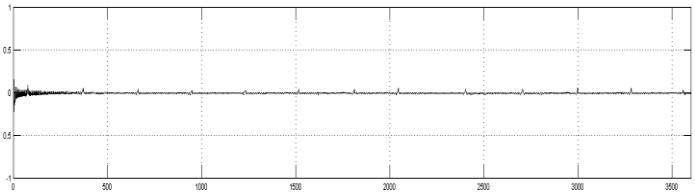
(d)



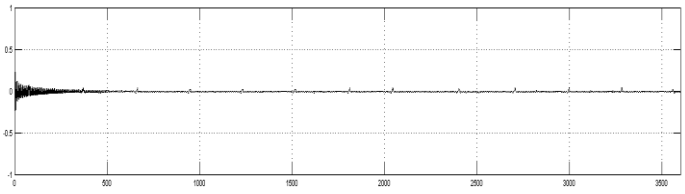
(e)



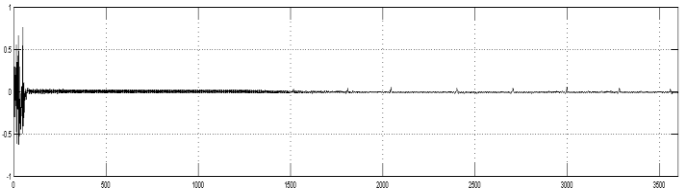
(f)



(g)



(h)



(i)

Fig.2. Simulation results for PLI noise removal: (a) Signal difference following AP filtering, (b) Signal difference following AP filtering designed with non-restoring division algorithm, (c) Signal difference following AP filtering designed using CORDIC algorithm, (d) Signal difference following RLS filtering with constant cross-correlation matrix (e) Signal difference following RLS filtering with constant cross-correlation matrix and designed using non-restoring division algorithm, (f) Signal difference following CORDIC based RLS filtering with constant cross-correlation matrix, (g) Signal difference following RLS filtering with update cross-correlation matrix, (h) Signal difference following non-restoring based RLS filtering with update cross-correlation matrix, (i) Signal difference following CORDIC based RLS filtering with update cross-correlation matrix.

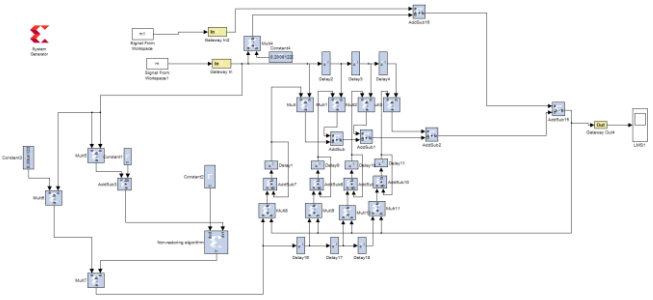


Fig. 3. Xilinx System Generator model of AP algorithm with non-restoring divider.

Table I presents the SNR comparison of various adaptive filters with different design metrics on the FPGA platform. These filters were tested for their effectiveness in removing PLI noise from a set of ten ECG signals obtained from the MIT-BIH database. The average SNR of the adaptive filter-based non-restoring algorithm is comparable to the average SNR ratio of adaptive filters with division terms performed outside the XSG tool. This demonstrates the robustness of the non-restoring algorithm in designing an adaptive filter for division tasks on the FPGA platform. In contrast, the CORDIC algorithm performs poorly in terms of SNR and convergence speed. Among the adaptive filters, the RLS adaptive filter with an updated cross-correlation matrix and a non-restoring divider exhibits the highest average SNR. The average SNR of the proposed RLS filter, incorporating an updated cross-correlation matrix, improved by 1.2% compared to the state-of-the-art RLS filter that incorporates a constant cross-correlation matrix, both utilizing the non-restoring division algorithm. The average SNR for AP, utilizing a non-restoring algorithm, improved by 6.4% compared to AP utilizing a CORDIC divider. The average SNR for RLS, incorporating a constant cross-correlation matrix, utilizing a non-restoring algorithm, improved by 16.6% compared to the same filter utilizing a CORDIC divider. Finally, the average SNR for RLS, incorporating an updated cross-correlation matrix, utilizing a non-restoring algorithm, improved by 37% compared to the same filter utilizing a CORDIC divider.

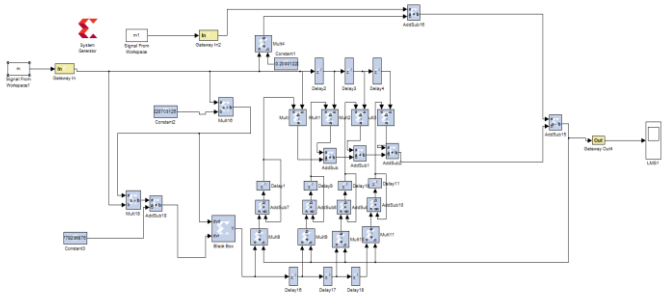


Fig.4. Xilinx System Generator model of RLS algorithm incorporates a constant cross-correlation matrix with non-restoring divider.

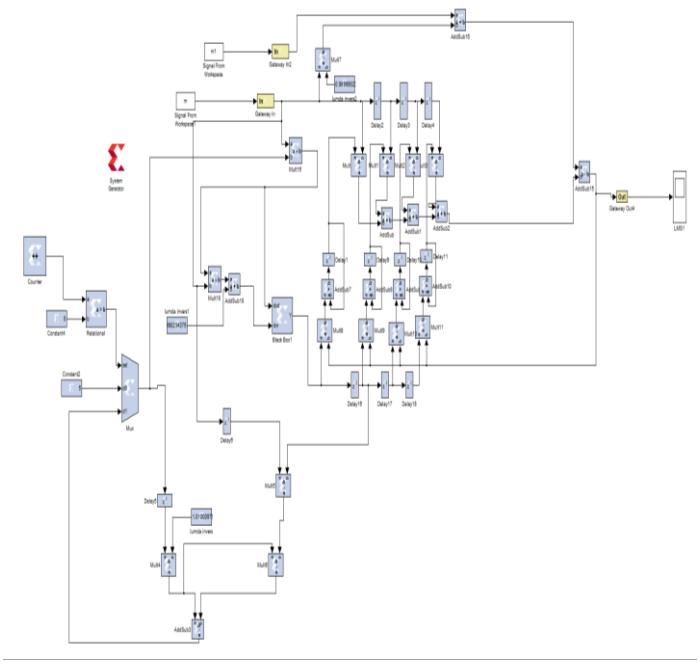


Fig.5. Proposed filter implementation of a four-tap RLS filter incorporates an updated cross-correlation matrix using a non-restoring divider.

Fig.6 depicts the convergence speed of various adaptive filters tested for removing PLI from ECG signal record 100. This figure provides valuable insights into the time taken by the filter to reach its steady state. The RLS adaptive filter, which incorporates an updated cross-correlation matrix, exhibits the highest convergence speed and the smallest steady state Mean Square Error (MSE). However, it should be noted that this algorithm lacks a true FPGA implementation and does not provide real resource utilization of the FPGA platform. Therefore, in this paper, it is only designed for comparison purposes. The convergence speed of the RLS filter with an updated cross-correlation matrix, along with the non-restoring division algorithm, is almost identical to that of the RLS filter with the MATLAB divider. This implies that the non-restoring division algorithm is the optimal choice for implementing adaptive filters on the FPGA platform. Furthermore, the convergence speeds of the AP and RLS filters designed with a constant cross-correlation matrix, using the MATLAB divider, are identical to those of the filters designed using the non-restoring division algorithm. On the other hand, the convergence speed of the RLS filters designed with the CORDIC division algorithm exhibits a large mean square error at the beginning. This is due to the fact that the CORDIC algorithm requires 31 cycles to initialize the division process, introducing an error during this period that increases the output error.

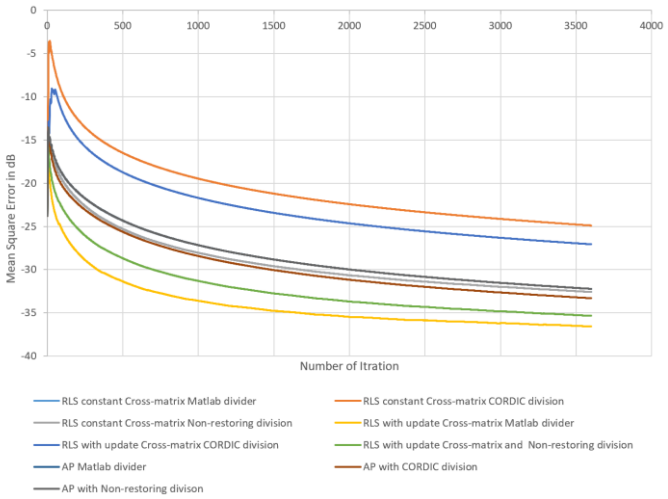


Fig.6. Convergence speed of various adaptive filters, the x-axis represents the mean square error of filter output, and the y-axis represents the number of iterations.

The resource utilization of the AP and RLS adaptive filters with the CORDIC and non-restoring division algorithms is presented in Table II. Comparing the two algorithms, the adaptive filter utilizing the non-restoring algorithm utilized fewer slice registers, but it required more LUTs and MUXCYs for implementation compared to the filter using the CORDIC algorithm. In terms of power consumption, the AP with the non-restoring division showed an 11% increase compared to the AP with the CORDIC divider. Similarly, the RLS filter with a constant cross-correlation matrix and a non-restoring divider consumed 8% more power than the RLS with a constant cross-correlation matrix using the CORDIC divider. Furthermore, when comparing the RLS filters with an updated cross-correlation matrix, the total power consumption increased by 33% for the non-restoring-based RLS compared to the CORDIC-based RLS. In summary, the non-restoring algorithm requires more resources and consumes more power when compared to the CORDIC divider.

TABLE I
SIGNAL-TO-NOISE RATIO CONTRAST OF VARIOUS ADAPTIVE FILTER ALGORITHMS IN DECIBELS

Record No.	AP Outside Xilinx division	AP designed Using non_ restoring	AP designed using CORDIC algorithm	RLS constant matrix Outside Xilinx division	RLS constant matrix designed using non-restoring	RLS constant matrix designed using CORDIC division	RLS update matrix Outside Xilinx division	RLS update matrix designed using non-restoring	RLS update matrix designed using CORDIC division
100	31.5931	31.5941	32.0515	30.9155	30.9138	31.1828	32.4081	32.6430	32.5170
101	31.6909	31.6910	32.1903	30.96	30.9626	23.2696	32.4379	32.6930	23.2290
102	30.2303	30.2313	30.8943	30.3511	30.3538	26.2186	32.4481	32.3981	13.5854
103	18.2842	18.2841	18.2913	16.5422	16.5426	16.3307	16.8919	17.1777	18.3832
104	27.0312	27.0321	27.1701	26.0467	26.0475	25.5847	26.6891	26.9599	14.6765
105	31.7972	31.7974	32.2974	31.0053	31.0057	15.2008	32.4338	32.6917	16.7669
108	31.5726	31.5752	32.0549	30.9698	30.9697	31.3673	32.5064	32.7485	12.6174
203	18.3168	18.3169	18.3103	16.5507	16.5509	16.0383	16.8930	17.4272	17.9474
220	24.5662	24.5658	4.4988	22.9980	22.9980	18.9830	15.2036	23.3087	23.3965
228	31.0055	31.0064	31.5725	30.7035	30.7034	24.6604	32.4475	32.5880	31.5866
Average	27.6088	27.60943	25.93314	26.70428	26.7048	22.88362	27.03594	28.06358	20.47059

TABLE II
RESOURCE UTILIZATION OF VARIOUS ADAPTIVE FILTER IMPLEMENT ON XILINX SPARTAN6 XC6SLX16-2CSG324.

Filter type and divider	Number of Slice Registers	Number of Slice LUTs	Number of occupied Slices	Number of MUXCYs used	No. of LUT Flip Flop pairs used	No. of Bounded IOBs	Power total in watt
AP with CORDIC divider	650	805	249	564	859	49	0.251
AP with non-restoring divider	160	1405	463	1156	1415	49	0.279
RLS with Constant cross-correlation and CORDIC divider	931	966	288	584	1054	49	0.273
RLS constant cross-correlation non-restoring	218	1450	471	1184	1505	49	0.297
RLS update cross-correlation CORDIC	1220	1072	315	620	1125	49	0.252
RLS update cross-correlation non-restoring	260	1794	625	1224	1855	49	0.336

CONCLUSION

This paper focused on designing an RLS filter using Xilinx System Generator (XSG) with an updated cross-correlation matrix to enhance the SNR and convergence speed of the RLS filter implemented with a constant cross-correlation matrix. This paper also proposed two different division algorithms to implement the RLS and AP filters on an FPGA platform. The results showed that the proposed filters with the non-restoring division algorithm outperformed the adaptive filters utilizing the CORDIC division algorithm in terms of SNR, convergence speed, and steady-state Mean Square Error. However, it should be noted that the proposed adaptive filters with the non-restoring division algorithm required more hardware resources and consumed more power when implemented on the FPGA platform compared to the filters using the CORDIC algorithm. Furthermore, the RLS and AP filters with the MATLAB divider exhibited the highest convergence speed and SNR. However, they lack a true implementation in the FPGA platform as they perform division operations outside of the FPGA. These designs were included in this paper for comparison purposes and system validation testing. The proposed RLS filter with the updated cross-correlation matrix and the non-restoring division algorithm showed promising results in terms of SNR and convergence speed. However, it is important to consider the trade-off between performance and resource utilization when selecting the appropriate division algorithm for FPGA implementation.

REFERENCES

[1] T. T. Hasan, M. H. Jasim, I. A. Hashim. FPGA Design and Hardware Implementation of Heart Disease Diagnosis System Based on NVG-RAM Classifier, 2018 Third Scientific Conference of Electrical Engineering (SCEE), 2018, pp. 33-38, http://doi.org/10.1109/SCEE.2018.8684125

[2] A. Z. Khan, I. Shafi . Removing Artifacts from Raw Electrocardiogram Signals Using Adaptive Filter in State Space. Circuits Syst Signal Process, 1713 (2020). http://doi.org/10.1007/s00034-019-01149-3

[3] T. M. Jamel, K. K.Al-Magazachi KK, 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, pp. 23-28.

[4] M. Kazemi, M. M. Arefi. A fast iterative recursive least squares algorithm for Wiener model identification of highly nonlinear systems, ISA Transactions, Volume 67, 2017, Pages 382-388, http://doi.org/10.1016/j.isatra.2016.12.002

[5] A. O. Abid Noor. Adaptive Noise Cancellation Using Noise Dependent Affine Projection Algorithm, Engineering and Technology Journal, Vol. 35, Part A, No. 6, 2017.

[6] G. Swaminathan, G. Murugesan, S. Sasikala, L. Murali. A novel implementation of combined systolic and folded architectures for adaptive filters in FPGA, Microprocessors and Microsystems, Volume 74, 2020, http://doi.org/10.1016/j.micpro.2020.103018

[7] M. Jayapravintha, S. Gomathi, G. Murugesan. Design of Systolic architecture for various adaptive filters for noise cancellation, 2015 3rd International Conference on Signal Processing, Communication and Networking (ICSCN), Chennai, India, 2015, pp. 1-6, http://doi.org/10.1109/ICSCN.2015.7219907

[8] V. Kavitha, P. K. Priya, Tha.Sugapriyaa. 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, pp. 171-177, Bangkok, 28-30 June, 2018, http://doi.org/10.18178/wcse.2018.06.031

[9] M. Chandra, P. Goel, A. Anand, A. Kar. Design and analysis of improved high-speed adaptive filter architectures for ECG signal denoising, Biomedical Signal Processing and Control, Volume 63, 2021, http://doi.org/10.1016/j.bspc.2020.102221

[10] S. C., S. S., M.G. Denoising ECG signal using combination of ENSLMS and ZA-LMS algorithms, 3rd International Conference on Signal Processing, Communication and Networking (ICSCN), 2015, pp. 1-6.

[11] S. Veni, "Real Time Implementation of SIGN LMS Adaptive Filters using Xilinx System Generator," International Journal of Mathematics and Computers in Simulation, vol. 14, 2020.

[12] S. Jayapoorani, D. Pandey, N. S. Sasirekha, et al, "Systolic optimized adaptive filter architecture designs for ECG noise cancellation by Vertex-5. AS 6," 163–173 (2023). http://doi.org/10.1007/s42401-022-00177-3

[13] F. Salehi, E. Farshidi, H. Kaabi. Novel design for a low-latency CORDIC algorithm for sine-cosine computation and its Implementation on FPGA, Microprocessors and Microsystems,Volume 77, 2020, 103197, ISSN 0141-9331, https://doi.org/10.1016/j.micpro.2020.103197

[14] R. S. Hongal, D. J. Anita. Comparative Study of Different Division Algorithms for Fixed and Floating Point Arithmetic Unit for Embedded Applications, 2016.

[15] F. Ding, Y. Wang, J. Ding. Recursive least squares parameter identification algorithms for systems with colored noise using the filtering technique and the auxilary model, Digital Signal Processing, Volume 37, 2015, Pages 100-108, http://doi.org/10.1016/j.dsp.2014.10.005

[16] Y. Hu. Iterative and recursive least squares estimation algorithms for moving average systems, Simulation Modelling Practice and Theory, Volume 34, 2013, Pages 12-19, http://doi.org/10.1016/j.simpat.2012.12.009

[17] D. Liu, H. Zhao.Affine Projection Sign Subband Adaptive Filter Algorithm With Unbiased Estimation Under System Identification, in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 70, no. 3, pp. 1209-1213, March 2023, http://doi.org/10.1109/TCSII.2022.3216807

[18] Y. Ren, Y. Zhi, J. Zhang. Geometric-algebra affine projection adaptive filter, EURASIP J. Adv. Signal Process. 2021, 82 (2021). http://doi.org/10.1186/s13634-021-00790-y