Taylor & Francis
Taylor & Francis Group

Automatika

Journal for Control, Measurement, Electronics, Computing and Communications

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/taut20

The design and implementation of folded adaptive lattice filter structures in FPGA for ECG signals

Kalamani C., Kamatchi S., Sasikala S. & Murali L.

To cite this article: Kalamani C., Kamatchi S., Sasikala S. & Murali L. (2023) The design and implementation of folded adaptive lattice filter structures in FPGA for ECG signals, Automatika, 64:4, 772-782, DOI: 10.1080/00051144.2023.2205725

To link to this article: https://doi.org/10.1080/00051144.2023.2205725

Published online: 30 May 2023.

AUTOMATIKA
2023, VOL. 64, NO. 4, 772–782
https://doi.org/10.1080/00051144.2023.2205725

REGULAR PAPER

OPEN ACCESS

The design and implementation of folded adaptive lattice filter structures in FPGA for ECG signals

C. Kalamani[a], S. Kamatchi[b], S. Sasikala[c] and L. Murali[d]

[a]Department of Electronics and Communication Engineering, Dr. Mahalingam College of Engineering and Technology, Pollachi, TN, India; [b]Department of Electronics and Communication Engineering, Amrita Vishwa Vidhyapeetham, Bangalore, India; [c]Department of Electronics and Communication Engineering, Kongu Engineering College, Perundurai, TN, India; [d]Department of Electronics and Communication Engineering, P.A. College of Engineering and Technology, Pollachi, TN, India

ABSTRACT

An adaptive filter is the utmost essential filter castoff in statistical signal dealing. The fine-tuning of the filter factor in relation to the response signal is the adaptive filter's key feature due to fewer calculations, Least Mean Square (LMS) adaptive filters are widely used to remove noise from Electrocardiograms (ECG). The adaptive filters are realized as signal processing algorithms in Digital Signal Processors (DSPs) or in VLSI Signal Processors (VSPs). The technique provides a way to create a folded adaptive lattice LMS filter, which requires less hardware than an adaptive lattice filter. Folding is an algorithm that uses a time scheduling technique that combines arithmetic operations into one operation which reduces Register and silicon chip areas. The design and implementation of a folded lattice adaptive filter remove Power Line Interference (PLI) noise from ECG signals. The MATLAB Xilinx System Generator tool is used to design the Adaptive Lattice LMS Filter and Folded Adaptive Lattice LMS Filter with Folding Order $K = 2$ and $K = 4$ and realized in the Virtex 5 FPGA KIT. The results of the folded architecture show that the area is reduced for $K = 2$ and $K = 4$ by 82.60% and 91.05%, respectively compared with a normal adaptive lattice filter.

ARTICLE HISTORY
Received 17 February 2023
Accepted 18 April 2023

KEYWORDS
Adaptive filter; folding; LMS; SNR; power line interference noise; Virtex5 FPGA

1. Introduction

In signal processing systems distortions of signal and noise are the main factor that limits performance. Practically, the method is working in a situation which has an uncertain input state where there are unexpected noises. Improvement in the system performance can be achieved by the removal of noise in the acquired signal, thereby increasing the Signal-to-Noise Ratio (SNR). Adaptive Noise Cancellation (ANC) is one of the main techniques which adjust their coefficient to diminish a fault signal. The fundamental benefit of this approach is that, unlike other signal processing methods, it is possible to obtain the degree of noise rejection with no priori predictions of signal or noise.

LMS and RLS are two adaptive noise cancellation methods. RLS algorithms use more sophisticated mathematical operations that need more computing power than LMS algorithms. LMS algorithm was considered for its less complexity and efficiency [1].

ECG signals play a critical role in the diagnosis of coronary diseases. During the acquisition, the ECG signal is contaminated by various other noises such as Power line Interference, baseline wander, muscular artefact and Electrode Motion noise [2]. To have a clear ECG signal, denoising is required. To make the denoising process easier, adaptive and efficient LMS algorithms have been reported in the literature. Realization of the LMS algorithm is incorporated as an infinite impulse response, lattice, transform domain filter and infinite impulse response. Among these types of realization, lattice arrangement is merged in the filter configuration to get better SNR and to achieve a reduction in delay. Folded architecture for various adaptive digital filters and its implementation in FPGA is reported in the literature.

2. Related works

In the literature, enormous work has been proposed in adaptive filters for the removal of noise in ECG. Some contributions particularly focusing on the VLSI architectures of adaptive filters for high speed, low power consumption and low area have also been proposed. In ref. [3], a combination of Systolic and Folding architectures has been proposed for various adaptive filters such as Recursive Least Square (RLS), Affine Projection (AP) and Kalman filters. In the paper [4] the folding technique for FIR filter has been used for area reduction. Adaptive LMS filter and Folded Adaptive LMS filter have been implemented in Xilinx in [5]. To reduce

CONTACT C. Kalamani kalamec18@gmail.com Assistant Professor, Department of Electronics and Communication Engineering, Dr. Mahalingam College of Engineering and Technology, Pollachi, Coimbatore 642003, TN, India

area and power consumption folded architecture for non-canonical Least Mean Square adaptive digital filters used in echo cancellation were implemented in ref. [6]. The design of the adaptive lattice LMS filter and its implementation has been done, which has an increased area. From the literature, it has been analysed that adaptive lattice LMS filters with the low area folding technique can be designed.

3. Adaptive lattice LMS filter

The adaptive filter uses the LMS method, which uses the "scheme of steepest descent" technique, and estimates the results continually by updating the filter's weights, as shown in Figure 1.

$$\text{Filter Result : } y(p) = w^T(p) \times (p) \tag{1}$$

$$\text{Error Message : } \; e(p) = d(p) - y(p) \tag{2}$$

Updated Weight Equation:

$$w(p+1) = w(p) + 2\mu e(p) \times (p) \tag{3}$$

The adaptive filter's components are depicted in Figure 1 and its operation is explained by the equation. The filter output, error signals and weight update equations are given in Equations (1)–(3), respectively. The fundamental latticework design that is used to create the adaptive system is depicted in Figure 2.

A Lattice filter is a very effective structure for error prediction using simultaneous forward and backward prediction mistakes. Each and all stages of the lattice are decoupled between each stage. For wide sense, stationary input data show orthogonal property identified from the backward prediction errors produced by the lattice predictor [7–10].

Because the lattice filter construction is modular, adding one or more stages is all that is necessary to improve the predictor's order without affecting earlier calculations. Every level of a lattice predictor has a similar structure, so if VLSI technology is seen to be advantageous for the application of interest, then such a predictor lends itself to its implementation.

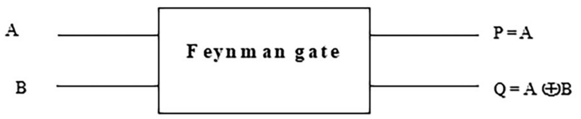


Figure 1. Configuration of an adaptive filter.

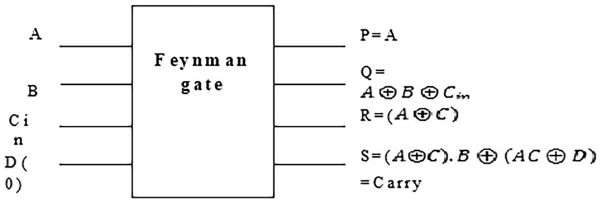

A
B
Cin
D(0)
Feynman gate
P=A
Q=$A \oplus B \oplus C_{in}$
R=$(A \oplus C)$
S=$(A \oplus C).B \oplus (AC \oplus D)$
=Carry

Figure 2. Basic lattice architecture.

Equation (4) specifies the forward error reduction update equation and Equation (5) gives the backward error order update equation.

$$f_m(p) = f_{m-1}(p) - k{*}_m(p) b_{m-1}(p-1) \tag{4}$$

$$b_m(p) = b_{m-1}(p-1) - k_m(p) f_{m-1}(p) \tag{5}$$

To transform a direct-form adaptive filter into a lattice filter, the lattice structure is shown in Figure 3. Figure 4 illustrates how the adaptive lattice LMS filter incorporates the LMS algorithm in addition to the lattice structure.

Figure 4 shows the 4-Tap Adaptive Lattice LMS Filter in which only the FIR section of the adaptive filter incorporates lattice structure.

4. Proposed folded adaptive lattice LMS filter

Folding is a technique that is incorporated into adaptive filters to reduce the area. Time multiplexed architecture with a reduction of silicon area is designed using the folding transformation technique. It is modified to time multiplex many algorithm operations into a single functional unit. It is employed to lessen the number of functional blocks. In a folded architecture, a functional unit is utilized again to perform N operations from the initial design in N unit times [11,12].

The folding transformation depicts an edge with a weight of D and a w(e) delay value extending from node U to node V. H_U is the functional source unit of node U pipelined by P_U through delays shown in Figure 5(a,b)

$$\mathrm{DG}_F(U \to V) = N_W(e) - P_U + v - u \tag{6}$$

4.1. Folding transformation technique for folding factor 2

A folding factor of $K = 2$ is chosen to design folded LMS adaptive Lattice filter. By splitting the LMS DFG [3,4] into two sections, its complexity is diminished. The Lattice filter block is in the upper portion as shown in Figure 6 and the lower portion is the filter weight adaptation block.

The lattice filter block's upper section is folded stepwise.

Step 1: Folding sequence $K = 2$ is selected and MP for multipliers and AD for adders folding sets given as

MP={MP0, MP1, MP2, MP3, MP4, MP5, MP6, MP7}

and

AD={AD0, AD1, AD2, AD3, AD4, AD5, AD6, AD7}.

Step 2: Folding Equation in Figure 7, the Data Flow Graph (DFG) is represented by the following folding equations.

$$\mathrm{DG}_F(S \to T) = N\, w\,(e) - P_S + t - s$$

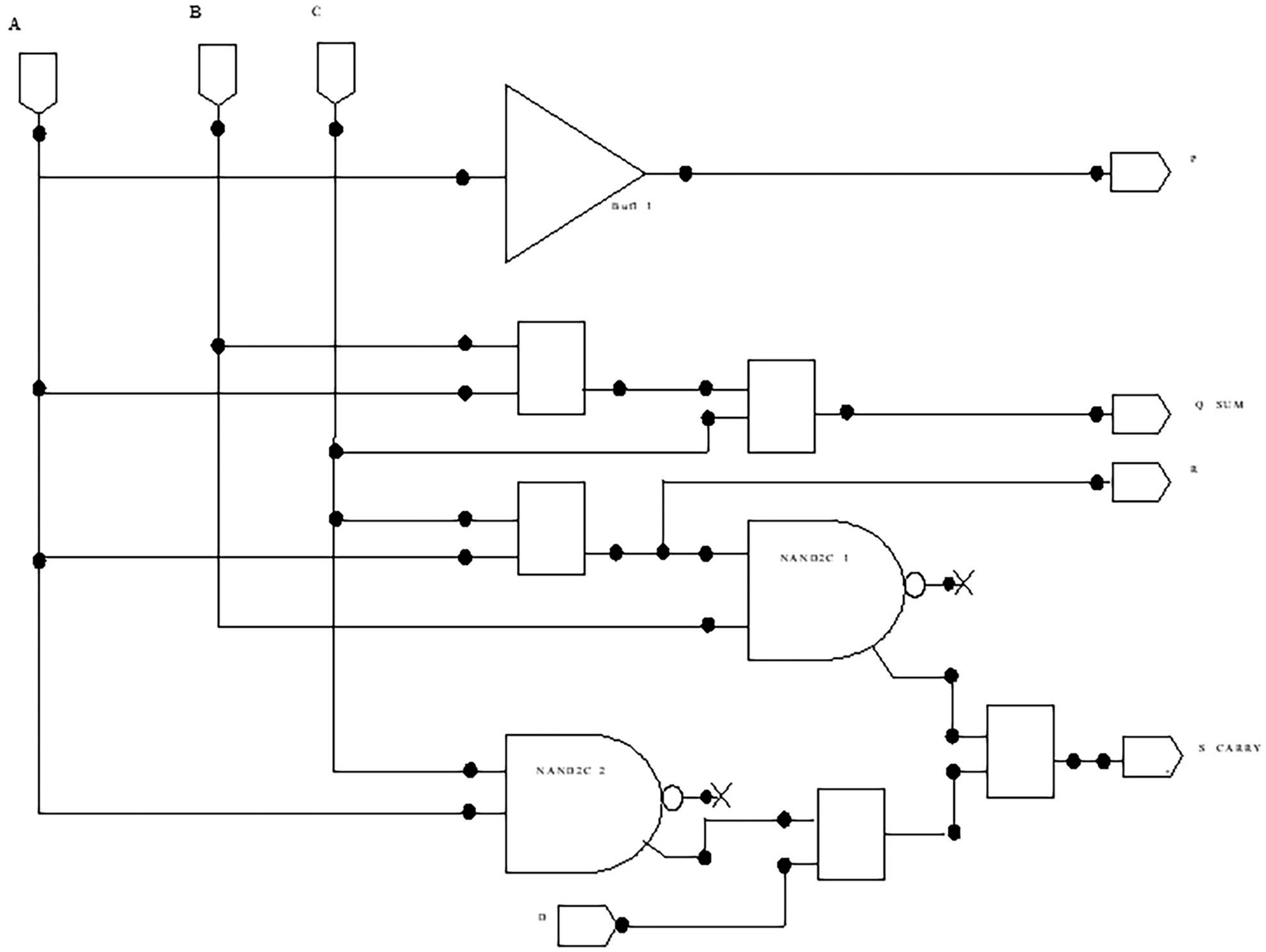


Figure 3. Lattice joint process estimator.

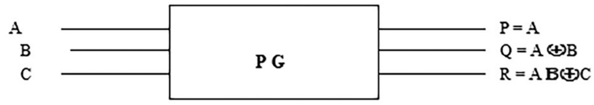


Figure 4. 4-Tap adaptive lattice LMS filter.

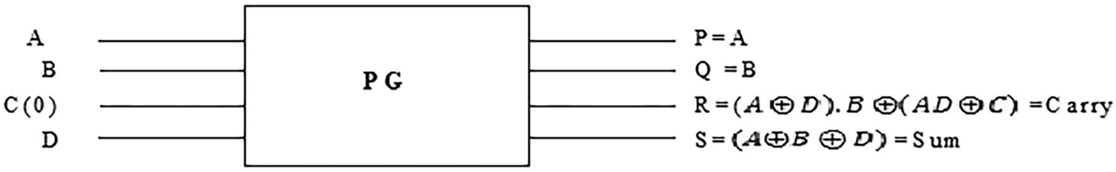


Figure 5. (a) DFG Node U to V with D Weight. (b) Corresponding folded edge.

$$\mathrm{DG}_F(\mathrm{MP0} \rightarrow \mathrm{AD0}) = 2*0-2+0-0=-2$$

$$\mathrm{DG}_F(\mathrm{MP1} \rightarrow \mathrm{AD1}) = 2*0-2+0-0=-2$$

$$\mathrm{DG}_F(\mathrm{MP2} \rightarrow \mathrm{AD2}) = 2*0-2+1-1=-2$$

$$\mathrm{DG}_F(\mathrm{MP3} \rightarrow \mathrm{AD3}) = 2*0-2+1-1=-2$$

$$\mathrm{DG}_F(\mathrm{MP4} \rightarrow \mathrm{AD4}) = 2*0-2+2-2=-2$$

$$\mathrm{DG}_F(\mathrm{MP5} \rightarrow \mathrm{AD5}) = 2*0-2+2-2=-2$$

$$\mathrm{DG}_F(\mathrm{MP6} \rightarrow \mathrm{AD6}) = 2*0-2+3-3=-2$$

$$\mathrm{DG}_F(\mathrm{MP7} \rightarrow \mathrm{AD7}) = 2*0-2+3-3=-2$$

$$\mathrm{DG}_F(\mathrm{AD1} \rightarrow \mathrm{AD3}) = 2*1-1+1-0= 2$$

$$\mathrm{DG}_F(\mathrm{AD3} \rightarrow \mathrm{AD5}) = 2*1-1+2-1= 2$$

$$\mathrm{DG}_F(\mathrm{AD5} \rightarrow \mathrm{AD7}) = 2*1-1+3-2= 2$$

$$\mathrm{DG}_F(\mathrm{AD0} \rightarrow \mathrm{AD2}) = 2*0-1+1-0= 0$$

$$\mathrm{DG}_F(\mathrm{AD2} \rightarrow \mathrm{AD4}) = 2*0-1+2-1= 0$$

$$\mathrm{DG}_F(\mathrm{AD4} \rightarrow \mathrm{AD6}) = 2*0-1+3-2= 0$$

$$\mathrm{DG}_F(\mathrm{AD0} \rightarrow \mathrm{MP3}) = 2*0-1+1-0= 0$$

$$\mathrm{DG}_F(\mathrm{AD1} \rightarrow \mathrm{MP2}) = 2*1-1+1-0= 2$$

$$\mathrm{DG}_F(\mathrm{AD2} \rightarrow \mathrm{MP5}) = 2*0-1+2-1= 0$$

$$\mathrm{DG}_F(\mathrm{AD3} \rightarrow \mathrm{MP4}) = 2*1-1+2-1= 2$$

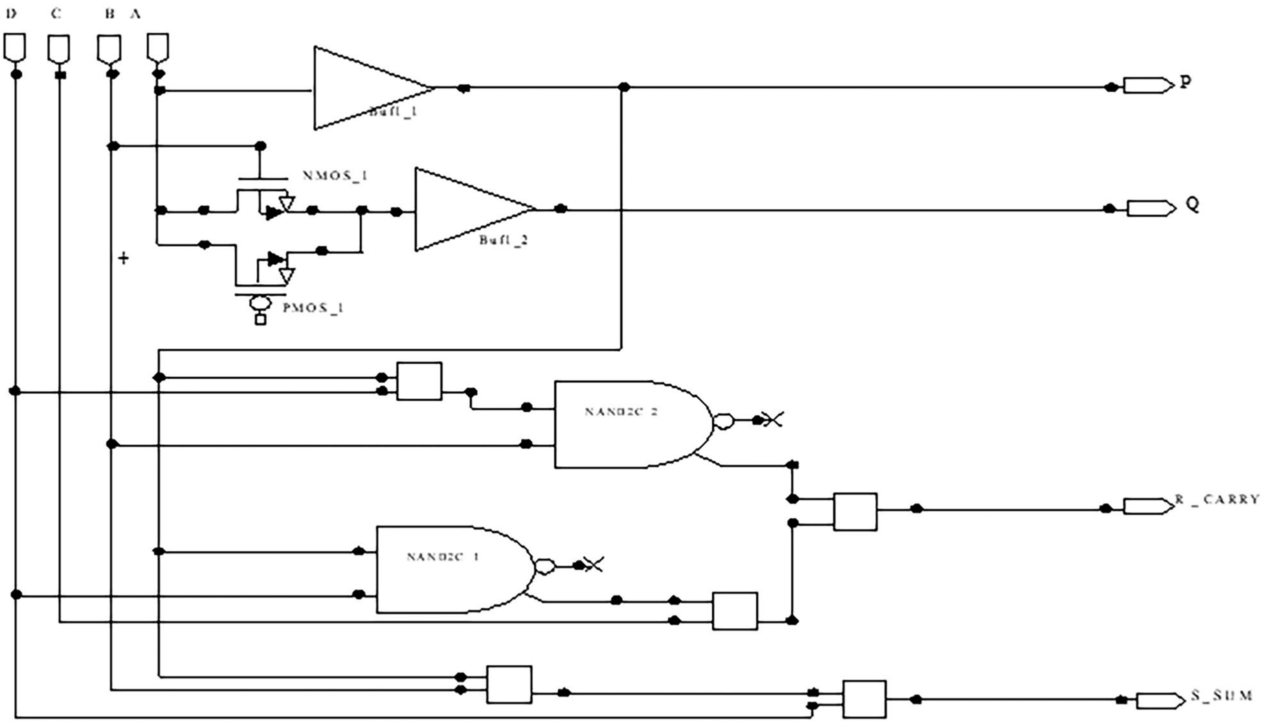


Figure 6. Lattice filter block.

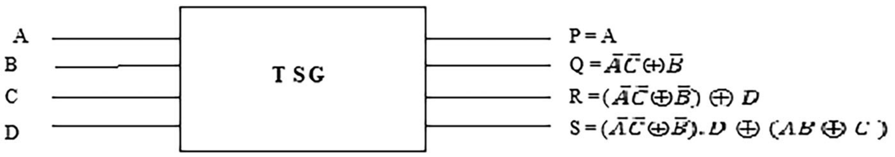

A
B
C
D
TSG
P = A
Q = $\bar{A}\bar{C}\oplus\bar{B}$
R = $(\bar{A}\bar{C}\oplus\bar{B})\oplus D$
S = $(\bar{A}\bar{C}\oplus\bar{B}).D\oplus(AB\oplus C)$

Figure 7. Lattice filter DFG with cutset pipelining.

$$\mathrm{DG}_F(\mathrm{AD4} \rightarrow \mathrm{MP7}) = 2*0-1+3-2 = \ 0$$

$$\mathrm{DG}_F(\mathrm{AD5} \rightarrow \mathrm{MP6}) = 2*1-1+3-2 = \ 2$$

Step 3: Perform Retiming for folding if needed

The DFG should not have a negative value to achieve proper folding. The negative delays on some edge are noted so retiming for folding is performed. The location of delay elements in the circuits is changed using the Retiming transformation technique. A special case of retiming known as Cut set [13,14] retiming is used in the design. DFG with the cut set is specified in Figure 7. The DFG after retiming is revealed in Figure 8.

Step 4: Construct a Lifetime table using the folding equations.

Table 1 displays the life duration for each node in the design, as determined by Equations (7) and (8).

$$\mathrm{Time}_{\mathrm{input}} = \mathrm{u} + \mathrm{Pu} \tag{7}$$

$$\mathrm{Time}_{\mathrm{output}} = \mathrm{u} + \mathrm{Pu} + \max \mathrm{v}\{\mathrm{DF}(\mathrm{U} \rightarrow \mathrm{V})\} \tag{8}$$

Table 1. Table of life period.

Node	Time$_{\mathrm{input}}$	Time$_{\mathrm{output}}$	Time$_{\mathrm{input}}$→Time$_{\mathrm{output}}$
MP0	$0+2=2$	$2+0=2$	2→2
MP1	$0+2=2$	$2+0=2$	2→2
MP2	$1+2=3$	$3+0=3$	3→3
MP3	$1+2=3$	$3+0=3$	3→3
MP4	$2+2=4$	$4+0=4$	4→4
MP5	$2+2=4$	$4+0=4$	4→4
MP6	$3+2=5$	$5+0=5$	5→5
MP7	$3+2=5$	$5+0=5$	5→5
AD0	$0+1=1$	$1+0=1$	1→1
AD1	$0+1=1$	$1+2=3$	1→3
AD2	$1+1=2$	$2+0=2$	2→2
AD3	$1+1=2$	$2+2=4$	2→4
AD4	$2+1=3$	$3+0=3$	3→3
AD5	$2+1=3$	$3+2=5$	3→5
AD6	–	–	–
AD7	–	–	–

Step 5: Lifetime chart construction and register count for the design

A chart for lifetime calculation for the Lattice structure is exposed in Table 2, which characterizes the

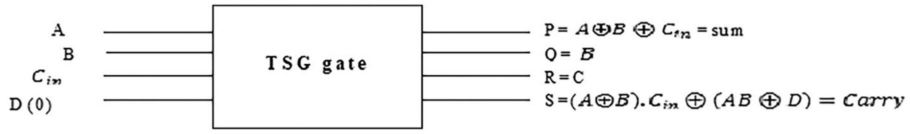

A
B
C_{in}
D (0)
TSG gate
P = $A\oplus B\oplus C_{in}$ = sum
Q = B
R = C
S = $(A\oplus B).C_{in}\oplus(AB\oplus D)$ = $Carry$

Figure 8. Retimed lattice filter DFG.

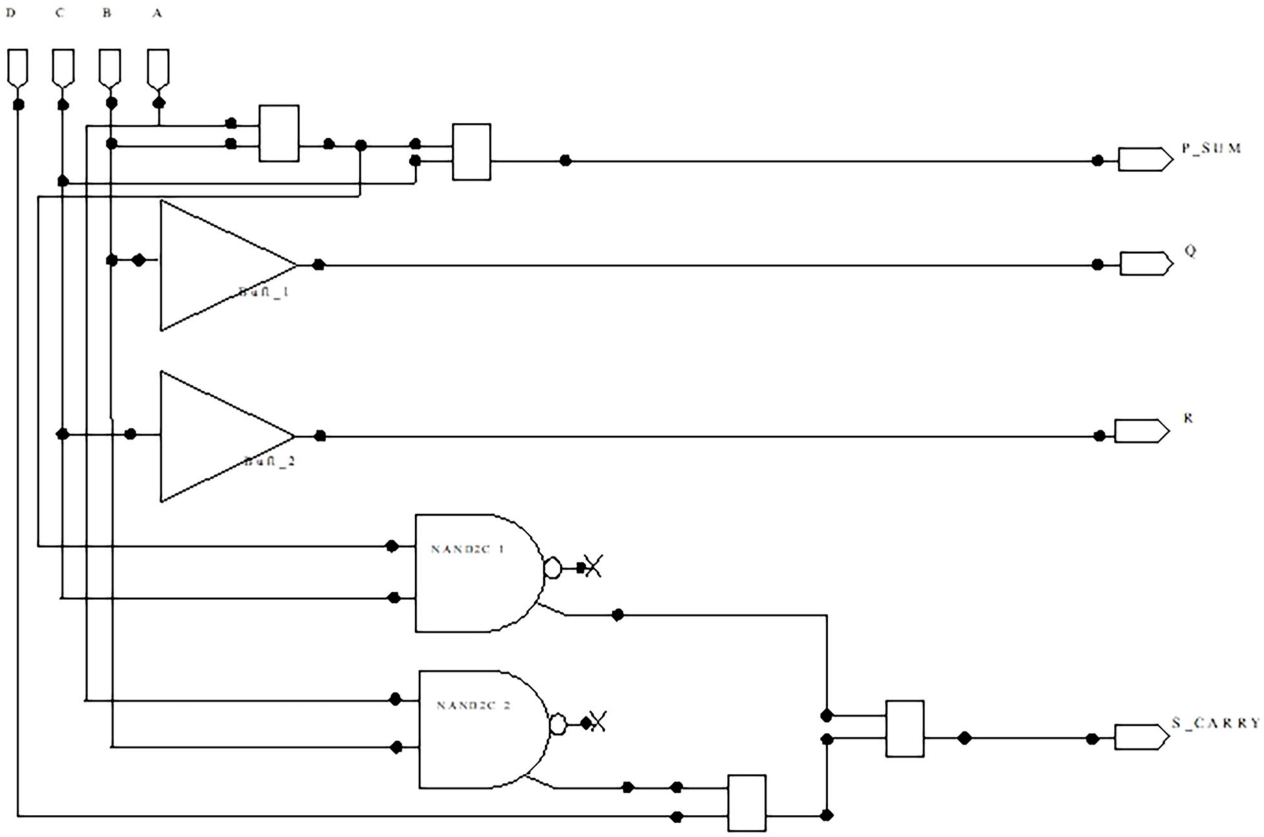


Figure 9. Folded lattice filter architecture of folding Order 2.

lifetime of every variable. The clock cycle is represented by horizontal lines and the lifetime of a variable is represented by vertical lines.

The bare minimum of registers required to achieve the design of a Lattice filter is obtained from the lifetime chart [15], given by the relation, the Minimum number of registers = Live variable maximum = max(0,1,2) = 2.

Step 6: Using the forward–backward register allocation approach, allocate data.

The registers are represented as RA1 and RA2 in the register allocation table shown in Table 3. Forward–backward allocation of registers is used for the variables. The registers are reused and the storage of dead variables is not done in this allocation technique.

Step 7: Allocation of Register.

The folded structure for the Lattice filter is created using all six of the aforementioned steps, as seen in Figure 9.

4.2. Folding transformation technique for order 4

The folding order is chosen as 4 and the above steps are followed for $K = 4$. Figure 10 shows the Folded Lattice filter architecture of folding order 4.

5. Results and discussion

The Xilinx System Generator Tool offers many Simulink blocks for hardware operations that can be implemented on Xilinx FPGAs. These simulink blocks are used in Simulink settings to functionally test the constructed system.

Utilizing Xilinx System Generator, the 8-tap Lattice LMS adaptive filter structure is developed and simulated. Figure 11 depicts an 8-Tap Lattice LMS adaptive filter without a folding design.

Figure 11 demonstrates the eight structures of the tap lattice LMS filter, Figure 12 denotes the 8-tap lattice LMS filter structure of folding order $K = 2$ and

Table 2. Plan of life time.

Cycle	MP0	MP1	MP2	MP3	MP4	MP5	MP6	MP7	AD0	AD1	AD2	AD3	AD4	AD5	# Live Variable
0															0
1									☆	↓					0
2(0)	☆	☆								↓	☆	↓			1
3(1)			☆	☆						☆		↓	☆	↓	2
4(0)					☆	☆						☆		↓	2
5(1)							☆	☆						☆	1

Table 3. Register allocation table for the lattice filter.

Cycle	Input	RA1	RA2	Output
0				
1	AD0,AD1			AD0
2(0)	MP0,MP1, AD2,AD3	AD1		MP0,MP1,AD2
3(1)	MP2,MP3, AD4,AD5	AD3	**AD1**	AD1,MP2,MP3, MP4,AD4
4(0)	MP4,MP5	AD5	**AD3**	MP4,MP5,AD3
5(1)	MP6,MP7		**AD5**	MP6,MP7,AD5

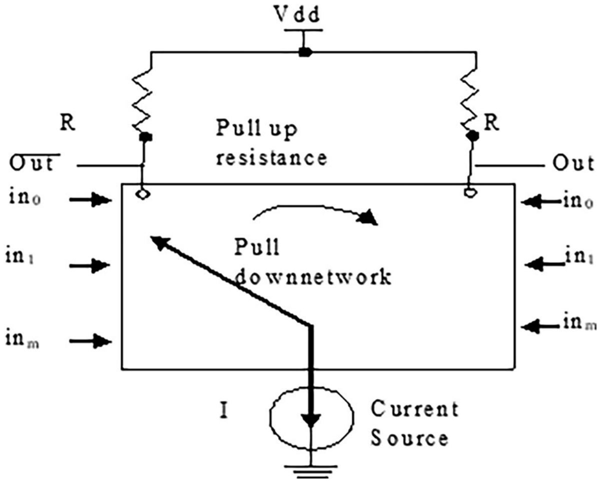


Figure 10. Folded lattice filter architecture of folding Order 4.

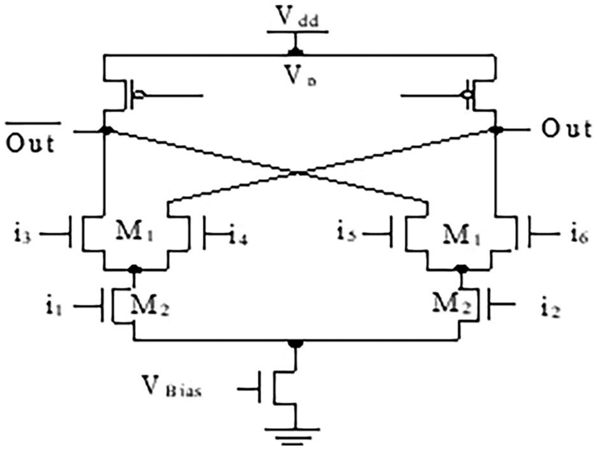


Figure 11. 8-Tap lattice LMS adaptive filter structure.

Figure 13 denotes that the 8-tap lattice LMS filter structure of folding order $K = 4$ is designed using a generator tool and simulated in Simulink background. Adder, multiplier, unit delay, division, math function, the constant, signal from a workspace and scope are the building components employed in this construction. Simulink blocks are connected to Xilinx blocks through the Gateway In, and Xilinx blocks are connected to

Simulink blocks through the Gateway Out. The design and simulation of 16-tap and 32-tap Lattice LMS Filters with folding orders $K = 2$ and $K = 4$ are similar.

5.1. Analysis of the proposed design

The input signal is added PLI noise at 50–60 Hz. The system generator is used for the simulation, while the FPGA design suite is used for the synthesis.

Model output of the 8-tap folded lattice LMS filter of order $K = 2$ is shown in Figure 14 and the 8-tap folded lattice LMS filter of order $K = 4$ is shown in Figure 15. In this simulation, the PLI noise is mixed with the ECG signal as the input, and the noise is then filtered using an 8-tap folded lattice adaptive filter to produce the output. The x-axis shows the number of samples, the y-axis the signal amplitude in mV, and the ECG signal (100.dat) taken from the database [16,17]. By deducting the adaptive filter output from the main input signal, the denoised ECG signal is produced.

5.2. Calculating the signal-to-noise ratio for ECG signals

When signal denoising, the Signal-to-Noise Ratio (SNR) is regarded as a crucial metric. It will change depending on the database signals used, and it would only be high if the noise % was decreased.

$$\text{SNR} = 10\log(\text{S}/\text{N}) \qquad (9)$$

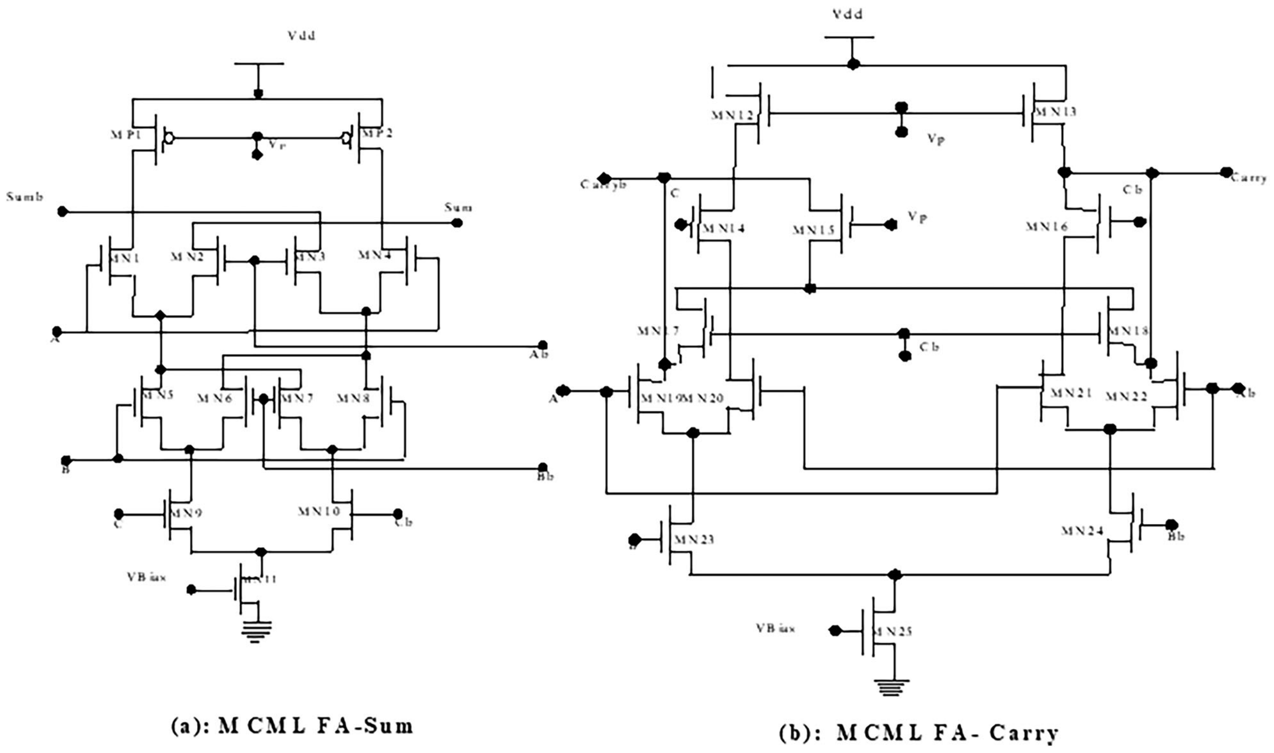
S – Signal in mV
N – Noise in mV

The comparison of SNR for the 8-Tap Lattice LMS filter and the 8-Tap Folded Lattice LMS filter of folding order 2 and order 4 is shown in Figure 16. It is witnessed that the SNR of the 8-tap folded lattice adaptive filter of order 2 and order 4 improved by 2.50% and 2.76% compared to the 8-tap lattice adaptive filter.

The comparison of SNR for the Lattice LMS filter and Folded Lattice LMS filter of folding order 2 and order 4 is shown in Figure 17. It is witnessed that the SNR of the 16-Tap folded lattice adaptive filter of order 2 and order 4 improved by 7.23% and 7.75% compared to the 16-Tap lattice adaptive filter. It is also perceived that the SNR of the 32-tap folded lattice adaptive filter of order 2 and order 4 improved by 27.83% and 29.82% compared to the 32-tap lattice adaptive filter.

5.3. Synthesis report

Table 4 shows the synthesis report of 4-, 8-, 16- and 32-Tap lattice normal, folded lattice LMS of order



(a): M CML FA-Sum

(b): M CML FA- Carry

Figure 12. 8-Tap folded lattice LMS adaptive filter structure of order ($K = 2$).

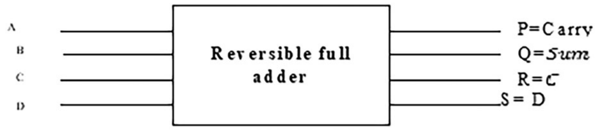


Figure 13. 8-Tap folded lattice LMS adaptive filter structure of order ($K = 4$) ECG signal generator in the Xilinx system.

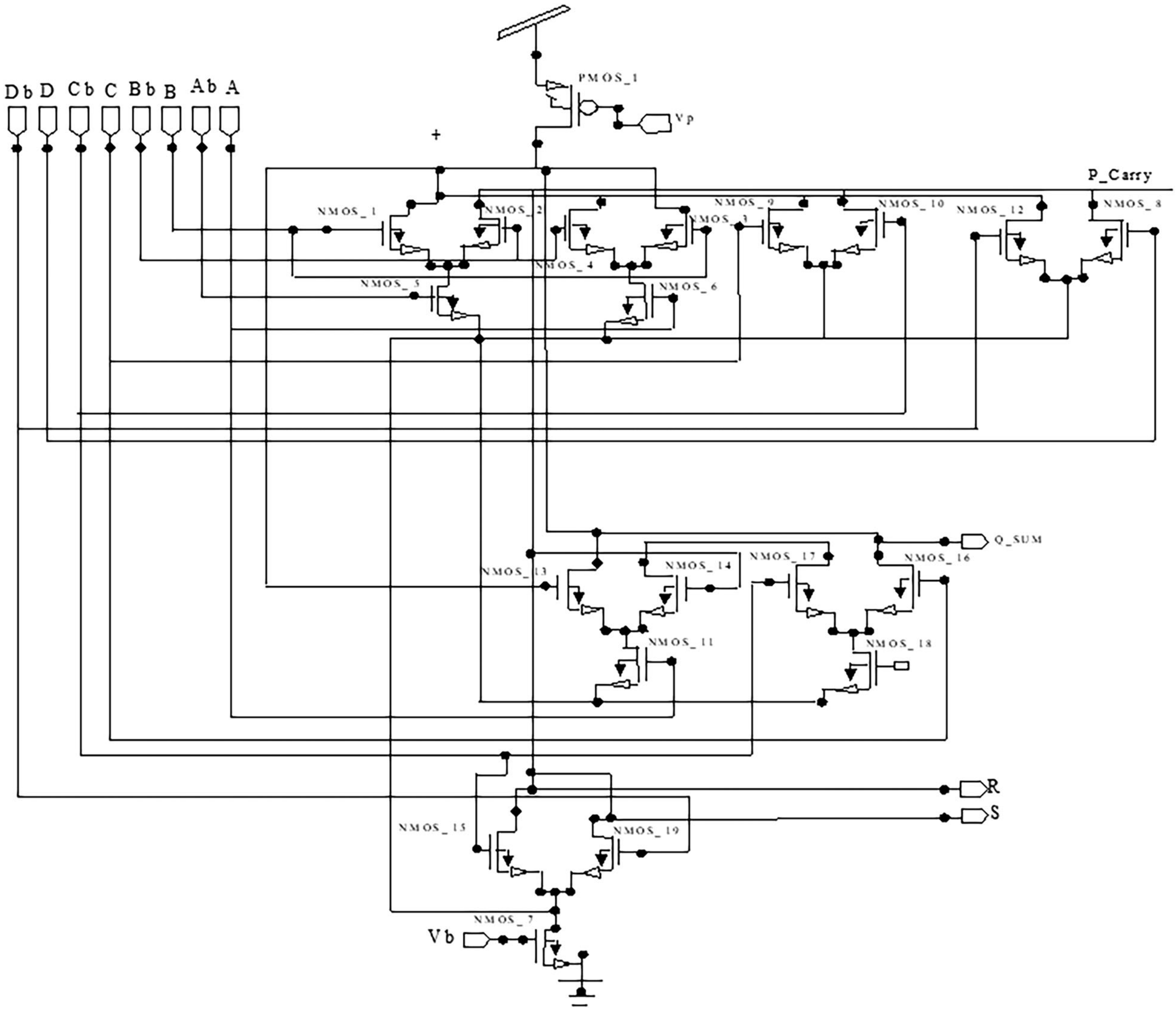


Figure 14. Simulation output of the 8-tap folded lattice LMS adaptive filter structure of order ($K = 2$) for 100.dat ECG signals.

Table 4. Synthesis report of the adaptive lattice filter and the folded adaptive lattice filter.

	4-Tap			8-Tap			16-Tap			32-Tap		
Filter/ parameter	Normal lattice filter	Lattice folding Order 2	Lattice folding Order 4	Lattice normal	Lattice folding Order 2	Lattice folding Order 4	Lattice normal	Lattice folding Order 2	Lattice folding Order 4	Lattice normal	Lattice folding Order 2	Lattice folding Order 4
Number of slice registers	177	33	17	369	65	33	753	129	65	1505	257	129
Number of slice LUTs	691	238	134	1315	446	238	2563	862	446	5007	1694	862
Number of LUT flip flop pairs used	738	248	143	1433	458	248	2824	878	458	5552	1718	878
Number of fully used LUT-FF pairs	130	23	8	251	53	23	492	113	53	960	233	113
Minimum input arrival time before clock (ns)	5.028	1.159	1.159	5.028	1.159	1.159	5.028	1.159	1.159	5.028	1.159	1.159
Maximum output required time after clock (ns)	0.807	0.471	0.471	0.807	0.471	0.471	0.807	0.471	0.471	0.807	0.471	0.471
Maximum combinational path delay (ns)	5.364	4.971	4.971	5.364	5.023	4.971	5.364	5.023	5.023	5.364	5.023	5.023

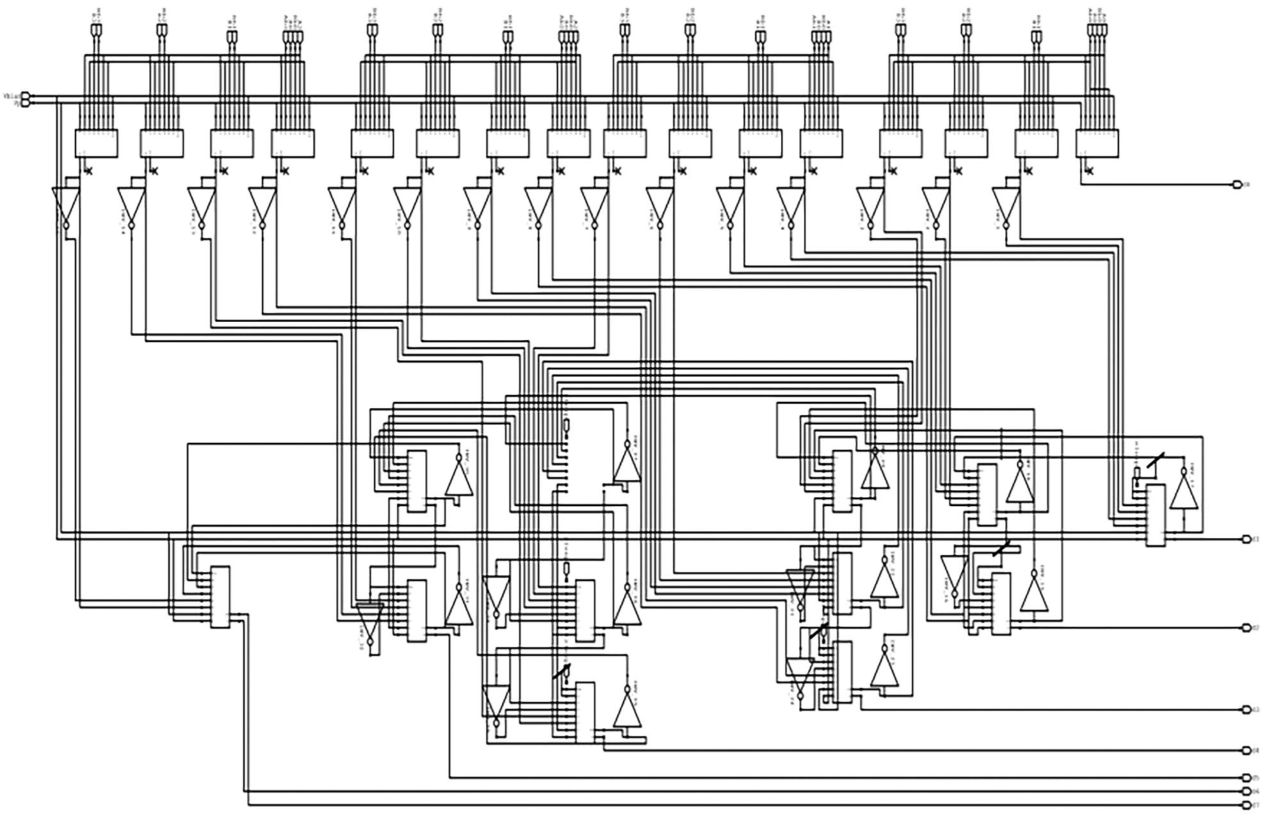


Figure 15. Simulation output of the 8-tap folded lattice LMS adaptive filter structure of order ($K = 4$) for 100.dat ECG signals.

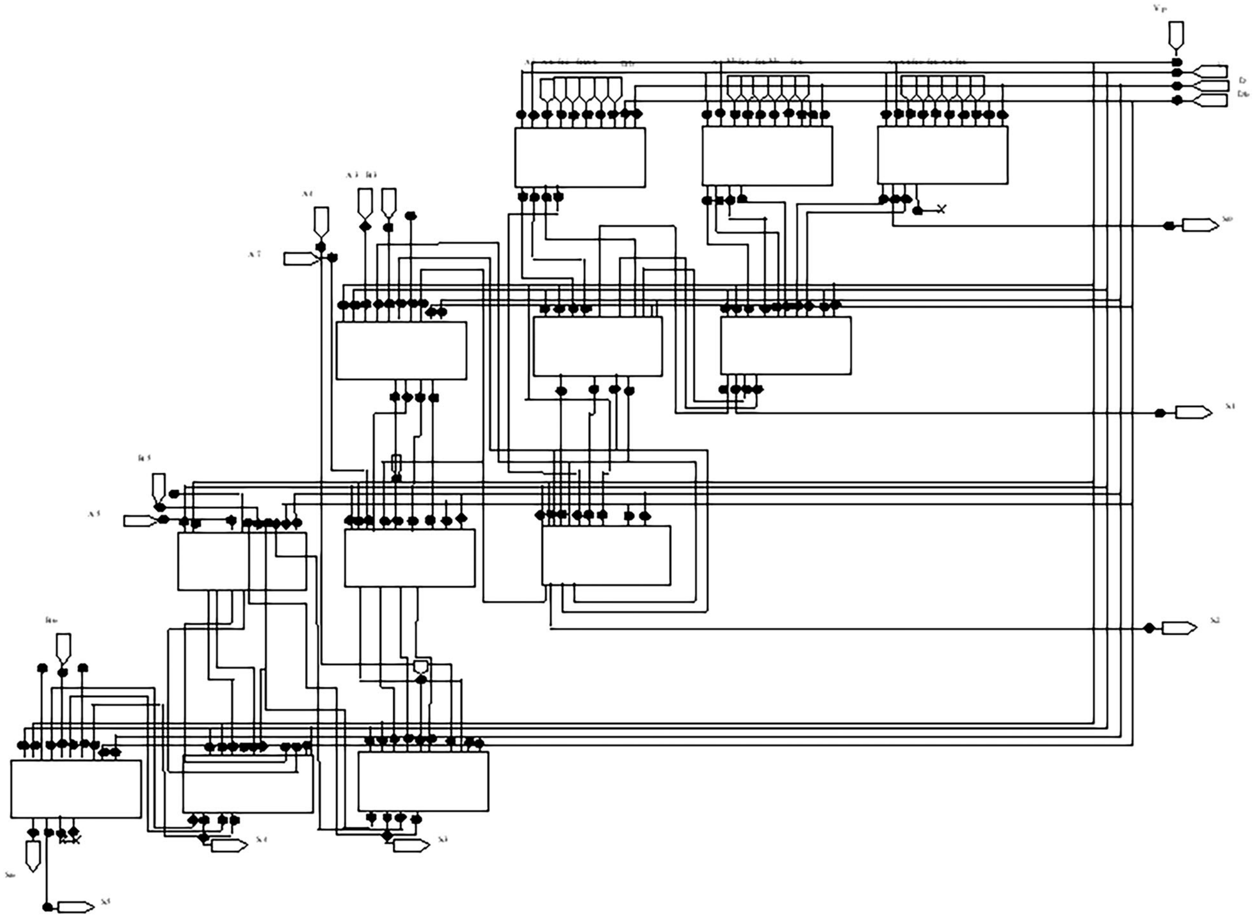


Figure 16. SNR of the 8-tap folded lattice LMS Filter.

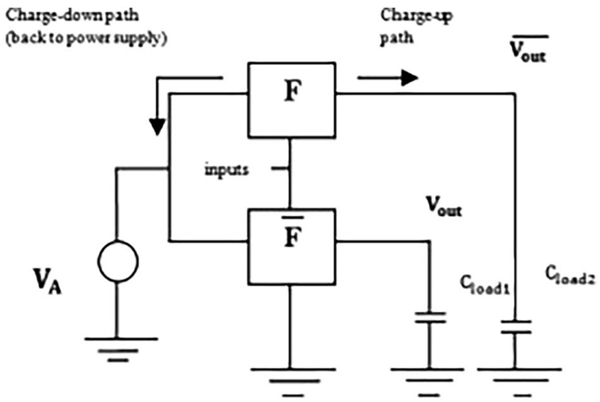


Figure 17. Comparison of SNR for different tap lengths of lattice filters and folded lattice filters.

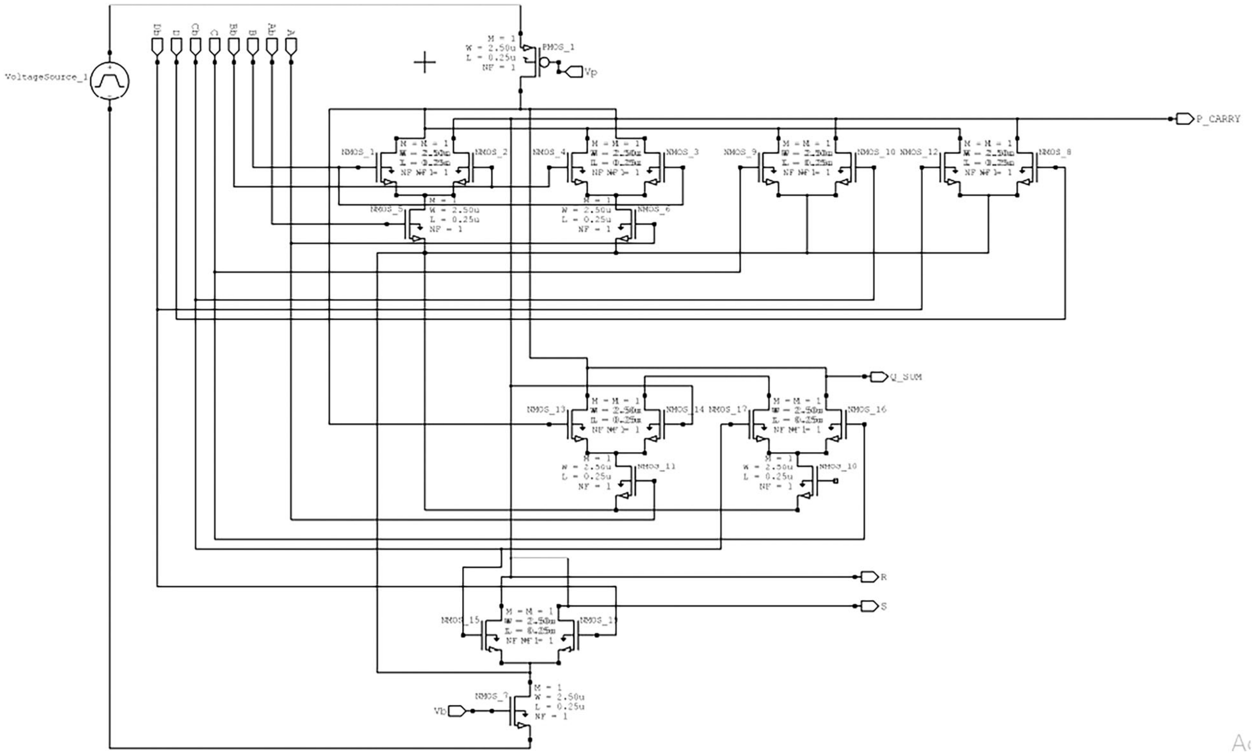


Figure 18. Comparison of device utilization report with different tap lengths of the lattice LMS filter with the folded lattice LMS filter.

$K = 2$ and folded lattice LMS of order $K = 4$ implemented using Virtex-5 FPGA.

The implementation results show that area of the folded architecture is reduced by 82.60% and 91.05% for $K = 2$ and $K = 4$, respectively compared to the normal adaptive lattice filter, as demonstrated in Figure 18.

6. Conclusion

Lattice LMS adaptive filters and lattice folded filters were created and used in the Xilinx system generator. Here, the Xilinx ISE tool was used to build folded adaptive lattice filters and lattice LMS adaptive filters in a digital context. The SNR of the suggested filters is calculated by simulation using various ECG signals from the MIT BIH database as input. The architectures are then implemented in Virtex-5 FPGA for analysis of area and speed. When compared to the standard adaptive lattice LMS filter, the suggested folded adaptive lattice LMS filters offer a reduction in hardware architecture. Using the implementation results it is observed that the area is reduced by 82.60% for folded adaptive lattice LMS filter of order $K = 2$ and 91.05% for folded adaptive lattice LMS filter of order $K = 4$.

Disclosure statement

No potential conflict of interest was reported by the author(s).

References

[1] Sasikala S, Murugesan G. Efficient digit serial architecture for sign least mean square adaptive filter for denoising of artifacts in ECG signals. Int J Biomed Eng Technol. 2017;23(2/3/4):335–344. doi:10.1504/IJBET.2017.082672.

[2] Sathya C, Sasikala S, Murugesan G. Denoising ECG signal using combination of ENSLMS and ZALMS algorithms. Proceedings of 3rd International Conference

on Signal Processing, Communication and Networking; August 2015. doi:10.1109/ICSCN.2015.7219911.

[3] Swaminathan G, Murugesan G, Sasikala S, et al. A novel implementation of combined systolic and folded architectures for adaptive filters in FPGA. Microprocess Microsyst. Feb 2020. doi:10.1016/j.micpro.2020.103018.

[4] Jayan G, Nair AK. Implementation of folded FIR filter based on pipelined multiplier array. 2018 3rd International Conference on Communication and Electronics Systems (ICCES); 2018 Oct 15–16. p. 667–672.

[5] Shanbhag NR, Parhi KK. Pipelined adaptive digital filters. Vol. 274. Springer Science & Business Media; 2012.

[6] Zode P, Deshmukh AY. Folded architecture for non canonical least mean square adaptive digital filter used in echo cancellation. Int J VLSI Des Commun Syst (VLSICS). 2016;7(3):34–39.

[7] Alsaid S, Abdel-Raheem E, Khalid M, et al. Variable tap-length algorithm using lattice structured LMS adaptive filters. IEEE 2017 30th Canadian Conference Electrical and Computer Engineering; 2017 Apr 30–May 3. p. 1–5.

[8] Patel S, Alex JSR. Optimized design platform for high speed digital filter using folding technique. Int J Res Electron Commu Technol. 2013;2(1):19–30.

[9] Sasikala S, Gomathi S, Naveen kumar D, et al. FPGA implementation of adaptive LMS lattice filter. Biosci Bitechnol Res Commun, Special Issue. 2020;13(3):121–125.

[10] Kalamani C, Pavithra N, Shanmathee S, et al. Design of efficient R electrocardiogram signal for electrocardiogram signal. Adv Eng Manage. 2021;9:111–127.

[11] Premkumar P, Kavitha S, Nandhini S. Design and implementation of folded FIR filter structures using high speed multipliers. Int J Adv Res Electr Electron Instrum Eng. 2014;3(12):13559–13565.

[12] Karuppuswamy R, Arumugam K. Folded architecture for digital gammatone filter used in speech processor of cochlear implant. ETRI J . 2013;35(4):697–705.

[13] Ayinala M, Brown M, Parhi KK. Pipelined parallel FFT architectures via folding transformation. IEEE Trans Very Large Scale Integr (VLSI) Syst. 2011;20(6):1068–1081.

[14] Jayapravintha M, Gomathi S, Murugesan G. Design of systolic architecture for various adaptive filters for noise cancellation. Proceedings of 3rd International Conference on Signal Processing, Communication and Networking; August 2015. doi:10.1109/ICSCN.2015.7219907.

[15] Maria viyani geno P, Sasikala S, Murugesan G. FPGA implementation of hybrid LMS adaptive filter using folded architecture. The conference in Connection with IETE South Zone Tech Fest 2K17 organized by IETE Centre at Kongu Engineering College, Perundurai; 2017.

[16] Singh S, Yadav KL. Performance evaluation of different adaptive filters for ECG signal processing. Int J Comput Eng. 2010;2(5):1880–1883.

[17] Islam SZ, Islam SZ, Jidin R, et al. Performance study of adaptive filtering algorithms for noise cancellation of ECG Signal. 2009 7th IEEE Conference on Information, Communications and Signal Processing (ICICS); 2009 Dec 8–10. p. 1–5.