Received: April 27, 2020. Revised: July 23, 2020. 156

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

VLSI Implementation of LFA based Median Filter with Noise Detection Architecture for EMG Denoising

Sharanabasappa[1]* Pallikonda Ravi Babu[2]

[1]*Department of Electronics and Communication Engineering, DonBosco Institute of Technology, India*
[2]*Department of Electrical and Electronics Engineering, Sreenidhi Institute of Science and Technology, India*
* Corresponding author's Email: sharan.k12@gmail.com

Abstract: Noise Reduction with less computational time and resources of the signal is a major requirement in signal processing. In the real-world scenario, Electromyography (EMG) signals often affect by various noises. The EMG signal requires noise reduction to obtain an accurate diagnosis process. In this study, Ladner-Fischer Adder (LFA) based Median Filter (MF) architecture is used to remove the noise from the EMG signal. The noise evaluation process is included in this proposed method to eliminate the flicker noise from the signal. The proposed LFA-MF method is simulated in the Cadence RTL compiler and Xilinx tool to evaluate the performance in terms of power, delay and Look Up Table (LUT). The objective of this work is to design the noise detection structure to avoid the denoising process for all the samples. This helps to reduce the hardware resources of the entire median filter architecture. The simulation result of LFA-MF architecture showed that it has lower power consumption and delay in noise reduction. The LFA-MF architecture has a power consumption of 1006548 nW in 16-bit sample width, while the existing method has 1537940 nW power consumption.

Keywords: Electromyography, Flicker noise, Ladner-fischer adder, Median filter, Xilinx.

1. Introduction

EMG is a bioelectric signal of the muscle tissue consisting of various features. The EMG signals often affect by noise that leads to lower accuracy in the diagnosis. Main noise sources are due to contaminates of the patient's electrodes are motion artifacts, baseline wander, and power line interference. EMG signal noise is difficult to detect and remove based on the linear filtering due to the non-stationary noises and overlap on the signals [1, 2]. Noise reduction is an important part of the analysis, diagnosis and other measurements of the signal in the medical domain. The median filter is a non-linear filter, which is used to remove the noise in the acquisition stage. This filter is also called as edge-preserving filter and it has various traits with reference to linear filters [3]. Median filter slices the entire signal into the window size and each window center value is replaced by the median value of the respective window [4]. Low-pass filters or Median filters or rank are commonly used for the digital filter noise removal in signal and image processing [5].

Field Programmable Gate Array (FPGA) depends on a parallel structure that can able to perform multiple processes simultaneously [6]. Transposed architecture in the high-speed filters provides more register efficient realization and the algorithmic delay elements have been used as pipelining register in the summation tree [7]. Median filters are a popular method for noise extraction and most research is based on the median filters to achieve high throughput and low hardware cost [8]. High-performance architecture of the median filter is in high demand for many applications. However, the median filter with large size has the limitations of high power consumption, low throughput and high area cost [9, 10]. The major contribution of this research work is given below:

- In this study, the LFA-MF architecture is implemented to reduce the noise in the EMG signal with less resource utilization.

- The noise evaluation module is applied in this architecture to detect noise in the input signal and the noisy signal is sent to LFA-MF. The LFA-MF is effectively reduced the noise in the binary data. The proposed LFA-MF method reduces the computation for noiseless data and compute noisy data.
- The architecture is simulated using the Cadence RTL compiler and Xilinx tool to analyse its efficiency. The simulation result shows that the LFA-MF architecture has lower power consumption compared to existing architecture.

This paper is organized as follows, the review of recent research in the noise reduction in signal is presented in Section 2, the LFA-MF architecture is discussed in the Section 3 and simulation result of architecture using two tools is presented in Section 4 and conclusion is drawn in Section 5.

2. Literature review

Noise removal in a signal based on hardware is required in processing of audio signals, EMG signal and Electrocardiogram (ECG) signals. The recent methods on hardware implementation of noise removal are reviewed in this section.

R.D. Chen, P.Y. Chen, and C.H. Yeh [11] designed a one-dimensional median filter with low power architecture for signal noise removal. The word-level pipeline filter was developed in two stages to receive an input signal and provide a median output at each cycle. The number of signal transition in the circuit was reduced to minimize power consumption. The token ring was used to set the stored samples in the window. The experimental results showed that power consumption was reduced in the developed median filter. The circuit area is high and needs to be reduced in the developed architecture.
C. Venkatesan, P. Karthigaikumar, and R. Varatharajan [12] studied a Delay Error Normalized Least Mean Square (DENLMS) for elimination of white Gaussian noise from the ECG signal. The delay elements were used to reduce the critical paths and pipelined VLSI architecture was used to improve the operational speed of adaptive filter. The experimental result of the developed DENLMS method showed that the power consumption was reduced very much. The cell leakage power was reduced effectively in this method. The circuit area of the developed architecture was increased due to the use of latches. The computational complexity of the developed method is slightly higher compared to the existing method.

L.A. Aranda, P. Reviriego and J.A. Maestro [13] developed a fault tolerant system based on the median filter and studied its performance. Protection method checks whether median output was within dynamic range or not, to develop the remaining non-median outputs. If the corrupted pixel was found in the system, then the output error signal was activated. Partial or complete reconfiguration was performed to remove the memory configuration error. The experimental result showed that the developed method can prevent 91 % of the corrupted image. The developed method has additional resource overhead for detection.

V. Kumar, A. Asati, and A. Gupta [14] designed a Low-Latency Median Filter (LLMF) in a 5×5 median filter to identify the median of 25 integer values. This method had the capacity of processing 25 integers in just three clock cycles. The architecture was made pipelined to improve the speed of the process. The three value sorter was used in this method that was faster than the conventional sorter. On the Xilinx Zynq FPGA device, the proposed method maximum frequency of operations was 394 MHz. The proposed method had a reduced clock cycle latency compared to the existing method. The processing time and response time of the developed method were reduced in the method. The power consumption of the developed method is high and the circuit area needs to be reduced in the architecture.
W.T. Chen, P.Y. Chen, Y.C. Hsiao, and S.H. Lin [15] developed a low cost and high throughput design for two dimensional (2D) hardware design. Parallel three value sorting methods and the functional sharing method were used to reduce the size of the circuit area. The number of pipelined was reduced under constant speed and the functional sharing method was used to implement median value sorting with the least number of comparisons. Two synthesized methods were used to develop the median filter. The experimental result showed that the area cost of the developed method was reduced up to 30% compared to existing methods. The power consumption of the developed method needs to be reduced and operational speed needs to be improved.

3. Proposed method

Electromyography (EMG) is the signal of electrical activities of muscles and the Motor Neurons (MN) for the diagnosis of neuromuscular diseases.

In this study, Ladner-Fischer adder (LFA) based Median Filter (MF) architecture is implemented to remove the noise from the EMG signal. The input EMG signal is first added to the Flicker Noise and

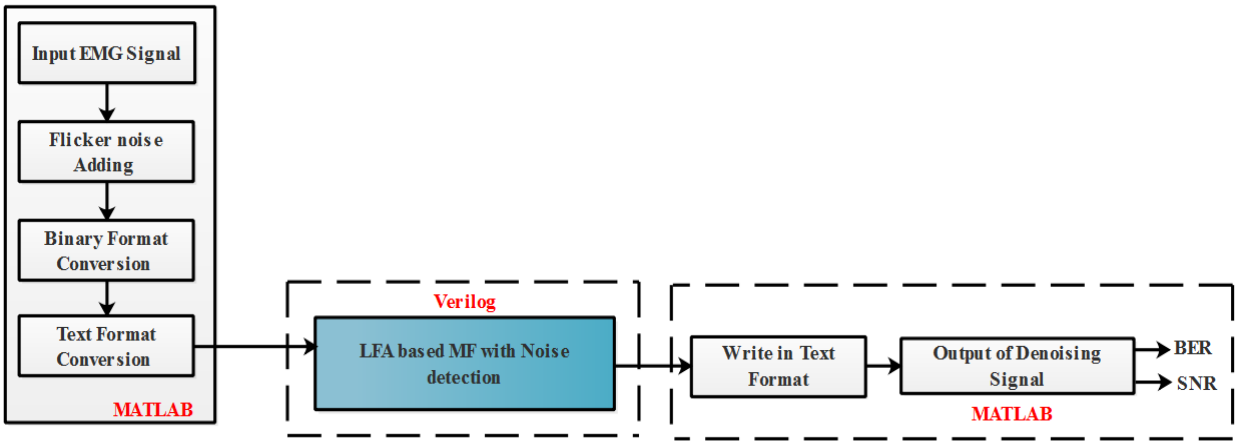


Figure. 1 The block diagram of the proposed method

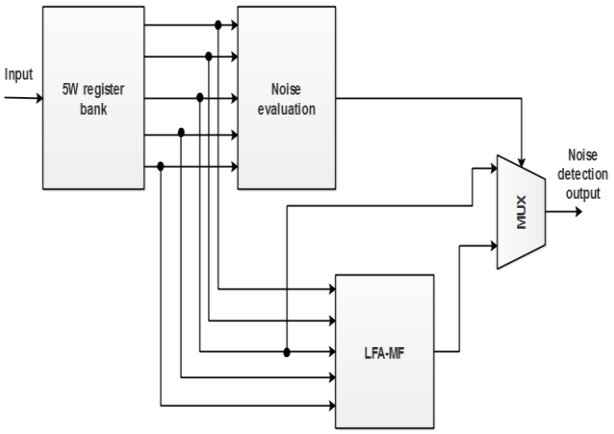


Figure. 2 Noise Evaluation module in the architecture

stored the noisy signal in the binary format. The binary format is converted into the text format to filter the noise. The LFA based MF architecture is used to detect and filter out the noise. The de-noised signals are stored in the text format and the output is analyzed with the parameters Bit Error Rate (BER) and Signal-to-Noise Ratio (SNR). The architecture based on LFA and MF is depicted in Fig. 1 and the noise evaluation module is presented in Fig. 2.

Text format conversion is given as input to the LFA based MF architecture to reduce the noise from the data. Input data is stored in the 5W register bank and the data is sent to the noise evaluation module and LFA-MF architecture. The noise evaluation module detects and provides information about

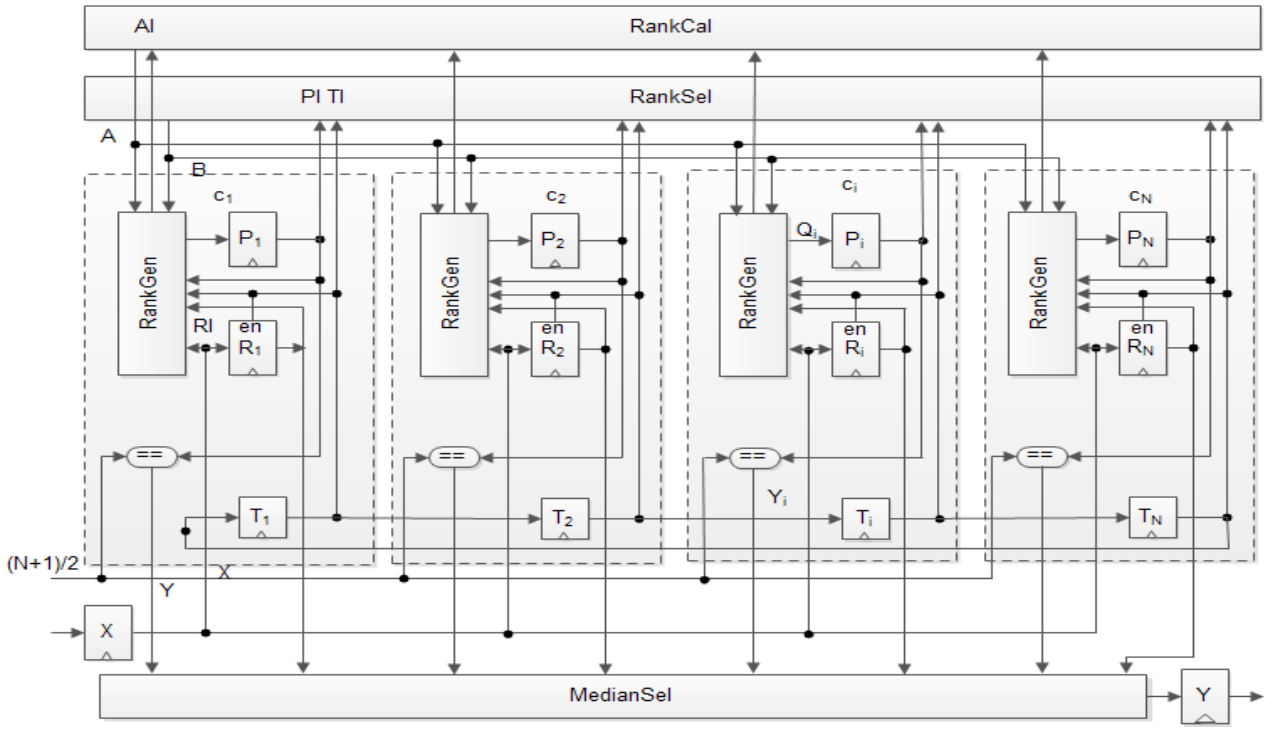


Figure. 3 low power 1D median filter architecture

whether the noise is present in the signal or not. Both the median value and the middle value are given to the input of the Multiplexer. Noise evaluation provides the output as 1 if noise is presented otherwise 0 is provided. The noise evaluation output is given as a selection line for the multiplexer. If selection line 1, the LFA-MF method reduces the noise in the signal otherwise the middle value is provided as output. This architecture reduces unwanted execution of noiseless data in the process.

3.1 Filter architecture

The proposed LFA-MF is applied in the Rank selection process for the noise reduction from signal. The hybrid of LFA and MF provides the effective noise reduction from the signal. Circuit design consists of three auxiliary modules Rank Selection (Rank Sel), Rank Calculation (Rank Cal) and Median Selection (Median Sel) and N identical cells, which is presented in Fig. 3. Modules are connected to the X input register and the median is stored in the Y output register. The growing edge of the total clock is used for the synchronization of the register architecture. The three registers Data Register (R_i), token register (T_i) and rank register (P_i) are present in each cell block c_i in the architecture. For instance, the sample cell c_i stores the data in register R_i, the sample rank keeps in P_i and the R_i, R_i permit signal is stored in T_i register. The rank is in the range of 1 to N for a cell based on a least sample value with the greatest example value in the N window size. For example, the sample value of R_i of cell c_i that consists of rank P_i is equal to $(N+1)/2$, where N is the odd number. This architecture provides the input based on the FIFO technique. Once the sample is queued, then it doesn't de-queued. If the sample has the token 1, then it helps to resist the other queue and remove the old samples simultaneously. Once the token is used, then it is provided to succeed in clock cycle.

In the first stage, the received sample is kept in the cell c_i, the previous cell is denoted as $c_N$and the shadow circle output is denoted as T_N. If the input sample arrives through the window, the rank cell gets updated. A detailed description of the circuit behaviour is provided in the research [11].

3.2 Circuit implementation

Fig. 4 presents the module of rank generation in a cell c_i. The input X_i and R_i is performed based on the " <= " operation that provides the output F_i. If $R_i > X_i$, then this provides the value of $F_i = 0$ else $F_i = 1$.

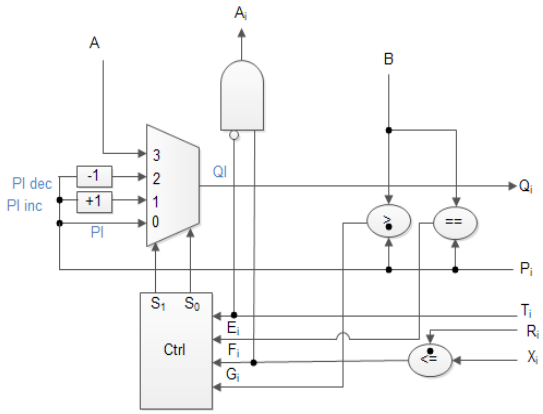


Figure. 4 Implementation of the rank gen module

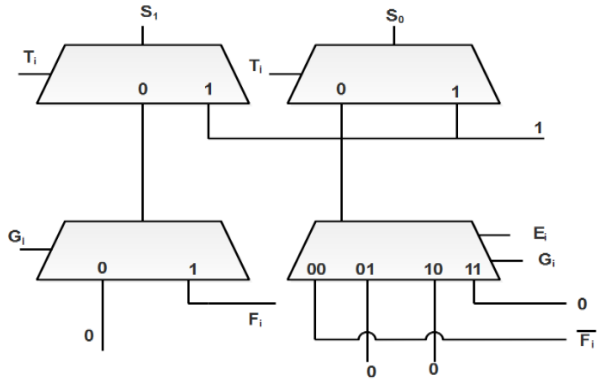


Figure. 5 Modified ctrl module

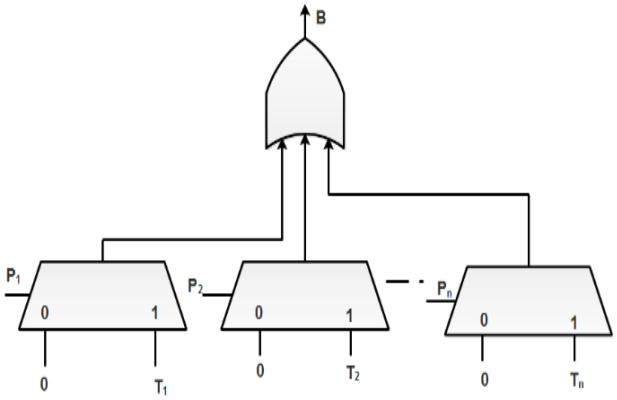


Figure. 6 Modified rank sel module

Fig. 4 presents the LFA is applied in +1 block. If $F_i = 1$ and $T_i = 0$, then AND gate provides the output of $A_i = 1$. If $A_i \leq R_i$, then the output of AND gate provides the value of $A_i = 0$ and the cell c_i doesn't hold the token. The signal A_i and Rank calculation is connected together, which is used to provide the fresh rank cell that stores the token. If $P_i > B$, this provides $G_i = 1$ else $G_i = 0$. Likewise, the P_i is equal to B that provides $E_i = 1$ else $E_i = 0$. If $R_i \leq X$, then output is $F_i = 1$ else $F_i = 0$.

Fig. 4 presents the four sources are given to the input of $4{:}1$ mux that provides the one resource signal Q_i. The two selection line (S_0, S_1) is generated

based on Ctrl module that is applied to control mux and measure the F_i, T_i, G_i and E_i four signals.

Fig. 4 presents four potential sources present to refresh the position of the cell c_i and a 4-to-1 multiplexer are used to select one of these sources for signal Q_i. The rank P_i is refreshed by the estimation of Q_i at each cycle. The multiplexer is constrained by two choice signals S_1 and S_0.

The Ctrl module is used these two signs to determine four signals T_i , E_i , F_i and G_i .The multiplexer (MUX) AND, XOR, and OR gates are used to reduce the power and delay of the system for effective use of the slices in FPGA. Fig. 5 presents the modified control units and these are used in the Rank Gen unit to assert the FPGA/Tool effectively. The area is reduced by optimum value and changing all the logic gates into Mux.

The developed method uses the modified RankSel that is useful in the area and power decreases. The proportionate multiplexers are used instead of the AND gates, that is presented in Fig. 6. The OR gate is used to consider the Hardware Complexity of the circuit.

The proposed LFA-MF method is applied instead of a normal adder with an LFA adder is used for the rank generation module that is presented in Fig. 6. The LFA adder performs the fast arithmetic process in various data processing methods. This adder is applied to reduce the area and power dissipation.

3.3 Ladner-fischer adder

The proposed LFA adder increases the speed of the binary addition and this looks like tree structure for high performance in arithmetic operations. The FPGA's are widely used in recent years due to the improvement in the speed of microprocessor-based applications like DSP and telecommunication. The developed LFA consists of two stages preprocessing and generation stage.

3.3.1. Pre-processing

In the preprocessing stage, generate and propagate are from each pair of inputs. The propagate provides "XOR" operation of input bits and generate provides the "AND" operation of input bits [7]. The propagate (P_i) and generate (G_i) are derived based on the Eq. (1) and (2).

$$P_i = A_i \; XOR \; B_i \tag{1}$$

$$G_i = A_i \; AND \; B_i \tag{2}$$

3.3.2. Generation stage

In the generation stage, the carry is generated for each bit is called as carry generate (C_g) and carry is propagated for each bit is carry propagate (C_p). The carry propagate and generate are used for further process and final cell present in each bit operate provide carry. The last bit carry provides the sum of next bit simultaneously. The carry generate and propagate are given in Eq. (3) and (4).

$$C_p = P_1 \; AND \; P_0 \tag{3}$$

$$C_g = G_1 \; OR \; (P_1 \; AND \; G_0) \tag{4}$$

The carry propagate and generate are presented in Eq. (3) and (4) is black cell and carry generation is presented in Eq. (5) is cell, i.e. gray cell. The carry propagate is used for further process and the final cell present in each bit operation provides carry. The last bit of carry leads to the sum of the next bit simultaneously. The carry is used for the next bit sum operate, the carry generate is presented in Eq. (5).

$$C_g = G_1 \; OR \; (P_1 \; AND \; G_0) \tag{5}$$

The first bit of carry $XORed$ with the next bit propagate, then the output is the sum. The two 3 bit addition operation is performed and each bit involves in pre-processing stage and generates stage, then it provides the final sum.

The first input bit goes in the pre-processing stage and produce propagate and generate. These propagate and generate undergoes generation stage that produce carry generates and carry propagates, then provides final sum. The block diagram of efficient LFA is depicted in Fig. 7.

The efficient LFA looks similar to the tree structure for high performance arithmetic operations and this is high speed adder that focuses on gate level

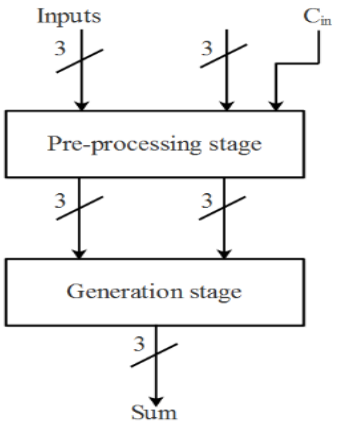


Figure. 7 The 3-bit LFA architecture

logic. This is designed with a reduced number of gates. This decreases the delay and memory used in the architecture.

4. Result and discussion

The proposed LFA-MF architecture is simulated on the Cadence RTL compiler 180 nm technology. The proposed LFA-MF architecture is evaluated in the FPGA Xilinx 14.4 ISE tool. The LFA-MF is analyzed in the system consists of Intel i5 processor with 8 GB of RAM and 500 GB hard disk. The architecture is tested on the 8-bit and 16-bit sample width and compared that with the existing method.

4.1 Tested on cadence RTL compiler 180nm

The proposed LFA-MF architecture is simulated in the cadence RTL compiler 180nm technology and analyzed with various metrics. The developed architecture is compared with the existing method [11]. The proposed LFA-MF architecture is evaluated in the 8-bit and 16-bit sample width and presented in Tables 1 and 2.

The proposed architecture is evaluated in terms of four parameters core area, power, delay, and EPS. The proposed method is compared with the existing method [11], LC-CLA-MF and MCRM-LC-CLA-MF. The output shows that the proposed LFA-MF architecture has lower core area, power, and delay compared to the existing method. The proposed LFA-MF method is tested on two window sizes and estimated efficiency. The proposed LFA-MF method has a delay of 2978 ps, while the existing method has a delay of 3014 ps in 5 window size.

The metrics ADP, APP and area value are measured for the proposed LFA-MF method and existing methods and presented in Figs. (8) -(10). The proposed LFA-MF method has higher efficiency compared to other existing methods. The reduction in area, power and delay of LFA-MF method are measured and presented in the Table 3. The proposed LFA-MF method has the considerable reduction in area, power and delay compared to the existing method. The LFA-MF method has the average of 11.33 % power reduction compared to the existing method.

Table 1. The area, power and delay of the proposed architecture in cadence RTL compiler on 8-bit sample width

Design	Throughput (#median outputs/clock)	Latency (#clock cycles)	Window size	8- bit sample width				
				Core area (um2)	Power (nW)	Delay (ps)	APP (area power product)	ADP (area delay product)
Existing [11]	1	w	5	96679	901943	3236	87198947297	312911251
			9	178585	2318289	4294	414011641065	766843990
LC-CLA-MF	1	w	5	48817	925495	2989	45179889415	145918894
			9	99895	2197765	4050	219545734675	404574750
MCRM-CLA-MF [16]	1	w	5	44178	869412	2415	38408883336	106689870
	1	w	9	96412	1960214	3845	188988152168	370704140
LFA-MF	1	w	5	38215	821678	2278	31400424770	87053770
	1	w	9	94381	1768216	3621	166885994296	341753601

Table 2. The area, power and delay of proposed architecture in cadence RTL compiler on 16-bit sample width

Design	Throughput (#median outputs/clock)	Latency (#clock cycles)	Window size	16- bit sample width				
				Core area (um2)	Power (nW)	Delay (ps)	APP (area power product)	ADP (area delay product)
Existing [11]	1	w	5	116928	1537940	3157	179828248320	369141696
			9	215565	3796961	4219	818491897965	909468735
LC-CLA-MF	1	w	5	69393	1535128	3157	106527137304	219073701
			9	137341	3879114	4219	532761395874	579441679
MCRM-LC-CLA-MF [16]	1	w	5	66478	1302478	3014	86586132484	200364692
			9	110487	3520012	4017	388915565844	443826279
LFA-MF	1	w	5	63248	1006548	2978	63662147904	188352544
			9	101421	3261244	3914	330758627724	396961794

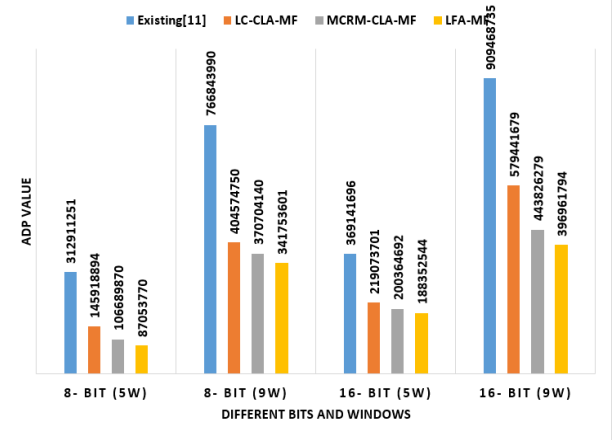


Figure. 8 The ADP value of the proposed LFA-MF method

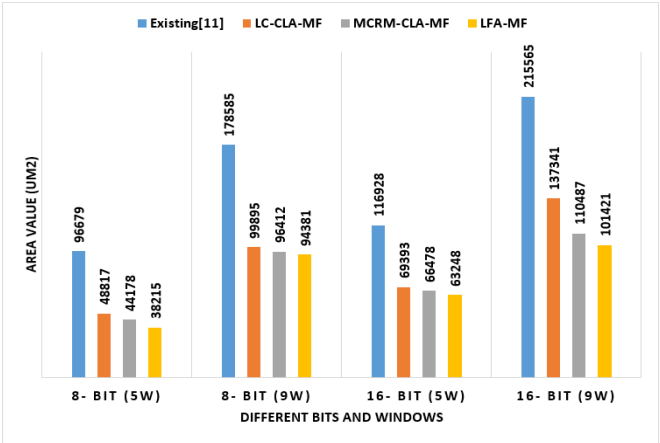


Figure. 10 The area value of the proposed LFA-MF method

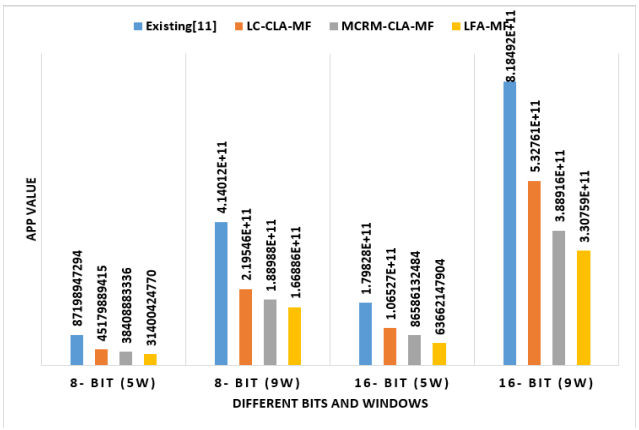


Figure. 9 The APP value of the proposed LFA-MF method

4.2 Testing on xilinx 14.4 ISE

The proposed LFA-MF architecture is tested using the FPGA Xilinx 14.4 ISE tool on 8-bit sample width and 16-bit sample width. The LUT, Flip flop and Slice are measured for the proposed LFA-MF method and existing method. The proposed LFA-MF method is evaluated using the FPGA Xilinx 14.4 ISE and compared it with the existing method.

The proposed LFA-MF method is evaluated for the two different windows (5 and 9). The LFA-MF method has less number LUT, Flip flops and slices compared to the existing method. The LFA-MF method has higher efficiency in both 8-bit and 16-bit length in FPGA analysis. The LFA-MF method is analyzed in 8-bit length is depicted in Table 4 and LFA-MF method is analyzed in 16-bit length is presented in Table 5. The LFA-MF method has the 104 LUT in the 5 window length in 16-bit length and the existing method has 120 LUT in 5 window length of 16-bit length.

The LFA-MF method is simulated and analyzed the LUT, Flip flops and slices which are presented in Figs. (11) -(13). The results point out that the proposed architecture has a less number of LUT, and flip flops compared to the existing method. The hardware resource requirement is reduced in the proposed architecture compared to existing method. The three types of FPGA design are used to evaluate the performance of the developed method. In Virtex 4, the proposed LFA-MF method has the 96 Flip flops in 5 window length in 16-bit sample width and the existing method requires 103 Flip flops in the architecture.

The median filter output waveform is presented in Fig. 14. The input value (X) of the waveform is considered as noise value.

Table 3. Reduction in area, power and delay of LFA-MF

Window	**Reduced % of Area**	**Reduced % of power**	**Reduced % of delay**	**Reduced % of APP**	**Reduced % of ADP**
5 –w (8 bit)	13.4	5.49	5.67	18.24	18.4
9 –w (8 bit)	2.10	9.79	5.82	11.69	7.8
5 –w (16 bit)	4.85	22.7	1.19	26.4	5.99
9–w (16 bit)	8.2	7.35	2.56	14.9	10.5
Average	7.1375	11.3325	3.81	17.8075	10.6725

Table 4. The proposed LFA-MF method is analysis on 8-bit sample width

8- bit sample width						
Target FPGA	**Circuit**		**LUT**	**Flip flop**	**Slice**	**Frequency (MHz)**
Virtex4 xc4vfx12	Existing [11]	5-w	165/10944	61/10944	110/5472	233.331
		9-w	321/10944	158/10944	188/5472	226.439
	LC-CLA MF	5-w	150/10944	61/10944	103/5472	234.995
		9-w	278/10944	109/10944	176/5472	184.795
	MCRM-CLA MF [16]	5-w	144/10944	61/10944	87/5472	231.25
		9-w	266/10944	109/10944	169/5472	184.795
	LFA-MF	5-w	127/10944	61/10944	72/5472	235.821
		9-w	253/10944	98/10944	158/5472	196.42
Virtex5 xc5vlx20T	Existing [11]	9-w	98/10944	109/10944	176/5472	184.795
		9-w	238/12480	130/12480	84/3120	259.96
	LC-CLA MF	5-w	71/12480	61/12480	25/3120	201.306
		9-w	180/12480	109/12480	67/3120	200.791
	MCRM-CLA MF [16]	5-w	74/12480	53/12480	23/3120	192.195
		9-w	135/12480	101/12480	40/3120	195.396
	LFA-MF	5-w	67/12480	45/12480	21/3120	196.193
		9-w	122/12480	106/12480	34/3120	196.244
Virtex6 xc6vcx75t	Existing [11]	5-w	120/46560	61 /46560	54 /11640	329.299
		9-w	185/46560	137/93120	90/11640	289.8
	LC-CLA MF	5-w	70/46560	61/93120	36/11640	251.49
		9-w	160/46560	109/93120	57/11640	242.404
	MCRM-CLA MF [16]	5-w	72/46560	53/93120	25/11640	247.878
		9-w	146/46560	102/93120	51/11640	242.055
	LFA-MF	5-w	68/46560	47/93120	22/11640	248.456
		9-w	141/46560	98/93120	47/11640	248.625

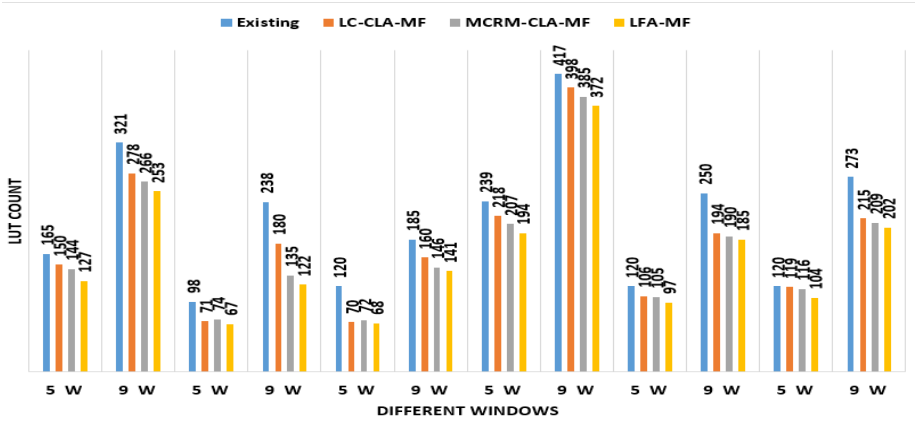


Figure. 11 LUT count for the various window size

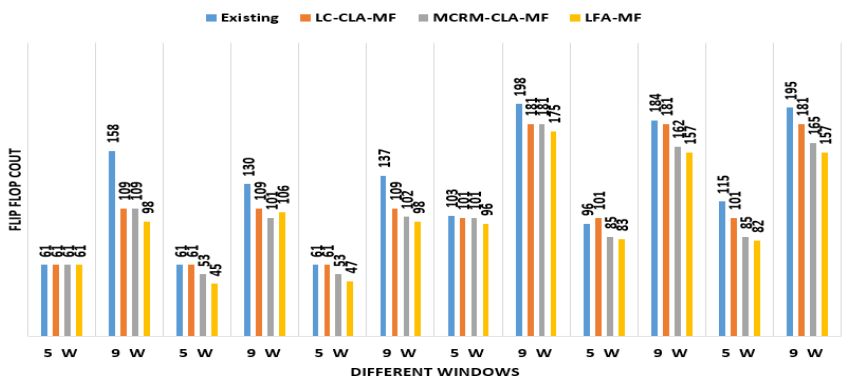


Figure. 12 FLIP FLOP count for the various window size

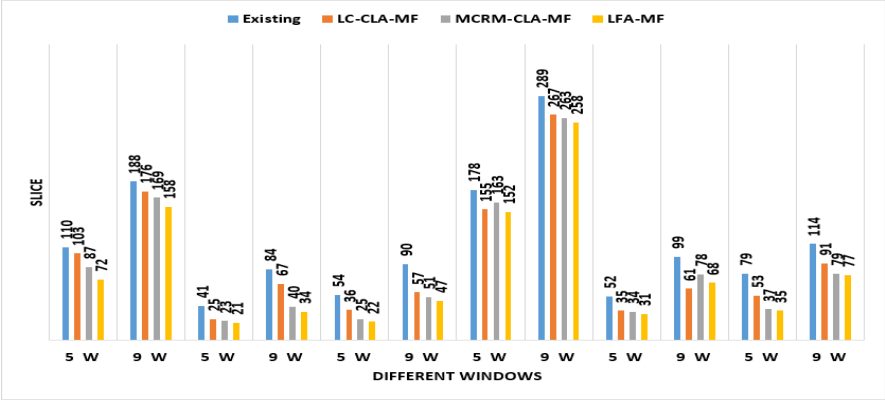


Figure. 13 Slice count for the various window size

Table 5. The proposed LFA-MF method is analysis on 16-bit sample width

16- bit sample width						
Target FPGA	**Circuit**		**LUT**	**Flip flop**	**Slice**	**Frequency (MHz)**
Virtex4 xc4vfx12	Existing [11]	5-w	239/10944	103/10944	178/5472	228.457
		9-w	417/10944	198/10944	289/5472	221.239
	LC-CLA MF	5-w	218/10944	101/10944	155/5472	223.584
		9-w	398/10944	181/10944	267/5472	175.951
	MCRM-LC-CLA MF [16]	5-w	207/10944	101/10944	163/5472	223.244
		9-w	385/10944	181/10944	263/5472	175.951
	LFA-MF	5-w	194/10944	96/10944	152/5472	225.124
		9-w	372/10944	175/10944	258/5472	178.148
Virtex5 xc5vlx20T	Existing [11]	5-w	120/12480	96/12480	52/3120	277.008
		9-w	250/12480	184/12480	99/3120	258.893
	LC-CLA MF	5-w	106/12480	101/12480	35/3120	277.335
		9-w	194/12480	181/12480	61/3120	185.147
	MCRM-LC-CLA MF [16]	5-w	105/12480	85/12480	34/3120	277.728
		9-w	190/12480	162/12480	78/3120	185.322
	LFA-MF	5-w	97/12480	83/12480	31/3120	285/164
		9-w	185/12480	157/12480	68/3120	192.458
Virtex6 xc6vcx75t	Existing [11]	5-w	120/46560	115/93120	79/11640	372.627
		9-w	273/46560	195/93120	114/11640	299.039
	LC-CLA MF	5-w	119/46560	101/93120	53/11640	305.022
		9-w	215/46560	181/93120	91/11640	236.616
	MCRM-LC-CLA MF [16]	5-w	116/46560	85/93120	37/11640	305.577
		9-w	209/46560	165/93120	79/11640	236.902
	LFA-MF	5-w	104/46560	82/93120	35/11640	306.258
		9-w	202/46560	157/93120	77/11640	248.125

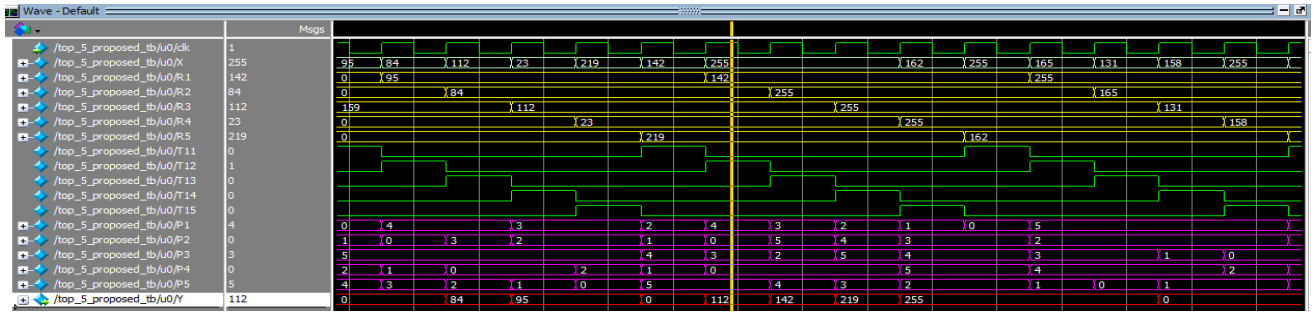
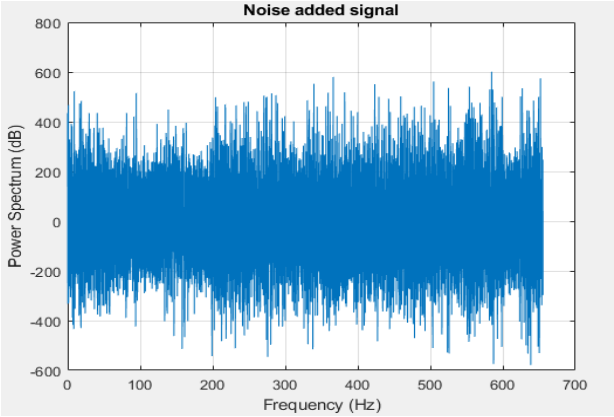
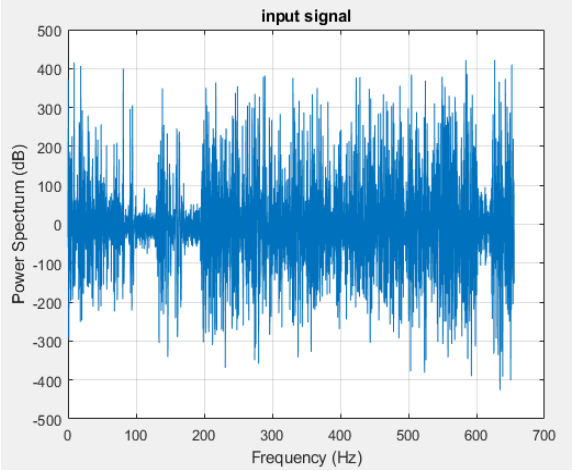


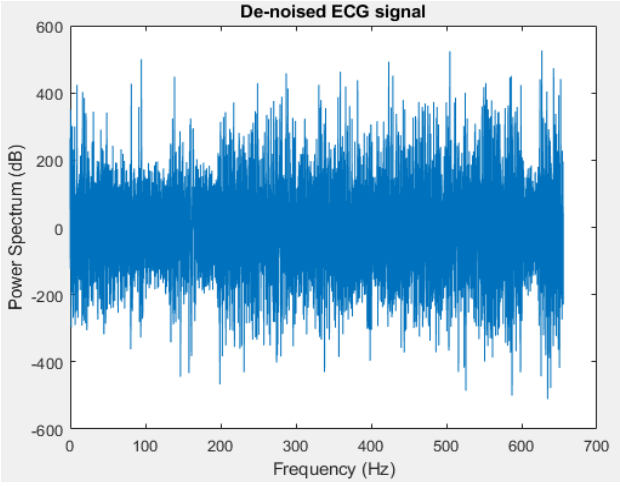
Figure. 14 The output waveform of median filter



(a)



(b)



(c)

Figure. 15 : (a) Input EMG signal, (b) noisy EMG signal, and (c) de-noised EMG signal

The noisy signal is converted into the binary value in MATLAB and that is provided to the Verilog. In the 5 window method, these input values are stored in 5 registers R1 to R5. The token value is represented as T11 to T15 and the rank value is mentioned as P1 to P5. Based on the perfect rank$(((N+1)/2)=3)$, the output is fetched from the register and stored in Y. The respective register $(R3)$ is provided to the output (Y).

Table 6. Comparative analysis of ASIC and FPGA

Cadence RTL 180nm	**Bit**	**Area**	**Power**	
LCMF [15]	8 bit	21254	36130000	
	16-bit	41940	66600000	
LFA-MF	8-bit	38215	821678	
	16-bit	63248	1006548	
Xilinx 14.4 ISE	**Bit**	**Flip Flop**	**LUT**	**Frequency**
LCMF [15]	8-bit	608	432	641.03
	16-bit	1184	864	613.5
LFA-MF	8-bit	47	68	248.45
	16 bit	82	104	306.25

Fig. 15 (a) presents the input signal of EMG signal. This input signal is red in MATLAB which undergone the flicker noise addition process. The noisy signal of EMG is depicted in Fig. 15 (b). The noisy data are converted to the binary values which is given to the input of the FPGA.

The MF is performed for denoising process and its output is stored as binary values. The output binary values are red in MATLAB to show the de-noised signal. The denoised EMG signal is depicted in Fig. 15 (c).

The proposed LFA-MF method is compared with Low cost Median Filter (LCMF) and comparison result is presented in Table 6. The proposed LFA-MF has consumed 821678 the power and existing method has consumed 36130000 power. The number of flip flops in proposed LFA-MF method is 47 and existing method is 608.

The comparative results of conventional method with proposed method is presented in Table 7. In this table, ENLMS [12], LLMF [14] and LCMF [15] methods takes more FPGA performances compared to the LFA-MF. From this table, it is clear that LFA-MF method performed effectively with less hardware utilization.

Table 7. Comparative results of FPGA

Design	**LUT**	**Flip flop**	**Slices**
ENLMS [12]	6998	5213	3867
LLMF[14]	1706	1550	2417
LCMF [15]	864	1184	687
LFA-MF	82	104	35

Table 8. BER, and MSE results for different signals

Types of Signal	**BER**	**MSE**
EMG signal [15]	0.1874	278.7886
ECG signal [16]	0.1743	248.5015
EMG signal	0.1642	201.4584

Once the signal is de-noised, the Mean Squared Error (MSE) and BER values are evaluated which is presented in Table 8. The proposed LFA-MF method and existing method are implemented in the same environment to analyse the performance. The proposed EMG signal has the MSE of 201.458, while the existing method has 248.5 MSE in the ECG signal.

5. Conclusion

Signal processing methods often require in the noise reduction process for accurate classification and this mostly requires for the less hardware usage. In this research, the LFA-MF architecture was proposed for the noise reduction process in the EMG signal. The EMG signal was added to Flicker noise to evaluate the efficiency of the proposed LFA-MF method. The noise evaluation module detected and provided the information about the noise in the signal. The module was used to reduce the noise reduction process in the noiseless signal. The proposed LFA-MF architecture was evaluated on the Cadence RTL compiler and Xilinx tool. The LFA-MF architecture was evaluated in 8-bit and 16-bit sample width. The simulation result showed that LFA-MF architecture had lower power consumption for noise reduction compared to the existing methods. The proposed LFA-MF architecture has the power consumption of 3261244 mW and existing method has the power consumption of 3520012 mW. In the future, the optimal two-dimensional median filter will be designed to remove the noises from Biomedical images and improves the FPGA performances.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

The paper conceptualization, methodology, software, validation, formal analysis, investigation, resources, data curation, writing-original draft preparation, writing-review and editing, visualization, have been done by 1st author. The supervision, and project administration, have been done by 2nd author.

References

[1] M. Marouf, L. Saranovac, and G. Vukomanovic, "Algorithm for EMG noise level approximation in ECG signals", *Biomedical Signal Processing and Control*, Vol. 34, pp. 158-165, 2017.

[2] J. Too, A. R. Abdullah, N. M. Saad, N. M. Ali, and T. N. S. T. Zawawi, "Exploring the Relation Between EMG Pattern Recognition and Sampling Rate Using Spectrogram", *Journal of Electrical Engineering & Technology*, Vol. 14, No. 2, pp. 947-953, 2019.

[3] M. Rahim and R. Maitheen, "Error Detection Technique for A Median Filter Using Denoising Algorithm", In: *Proc. of International Conf. on ISMAC in Computational Vision and Bio-Engineering*, pp. 1261-1270, 2018.

[4] V. G. Moshnyaga and K. Hashimoto, "An efficient implementation of 1-D median filter", *IEEE 52nd International Midwest Symposium on Circuits and Systems*, pp. 451-454, 2009.

[5] D. Prokin and M. Prokin, "Low hardware complexity pipelined rank filter", *IEEE Transactions on Circuits and Systems II: Express Briefs*, Vol. 57, No. 6, pp. 446-450, 2010.

[6] M.L. Zhu and D.Y. Ge, "Image quality assessment based on deep learning with FPGA implementation", *Signal Processing: Image Communication*, Vol. 83, pp. 115780, 2020.

[7] A. Blad and O. Gustafsson, "Bit-level optimized FIR filter architectures for high-speed decimation applications", In: *Proc. of IEEE International Symposium on Circuits and Systems*, pp. 1914-1917, 2008.

[8] A. Sanny and V.K. Prasanna, "Energy-efficient median filter on FPGA", In: *Proc. of International Conf. on Reconfigurable Computing and FPGAs (ReConFig)*, pp. 1-8, 2013.

[9] S. H. Lin, P. Y. Chen, and C. H. Lin, "Hardware design of an energy-efficient high-throughput median filter", *IEEE Transactions on Circuits and Systems II: Express Briefs*, Vol. 65, No. 11, pp. 1728-1732, 2018.

[10] C. Lin, W. T. Chen, Y. C. Chou, and P. Y. Chen, "A Novel Comparison-Free 1D Median Filter", *IEEE Transactions on Circuits and Systems II: Express Briefs*, pp. 1-1, 2019.

[11] R. D. Chen, P. Y. Chen, and C. H. Yeh, "A low-power architecture for the design of a one-dimensional median filter", *IEEE Transactions on Circuits and Systems II: Express Briefs*, Vol. 62, No. 3, pp.266-270, 2014.

[12] C. Venkatesan, P. Karthigaikumar, and R. Varatharajan, "FPGA implementation of modified error normalized LMS adaptive filter for ECG noise removal", *Cluster Computing*, Vol. 22, No. 5, pp. 12233-12241, 2019.

[13] L. A. Aranda, P. Reviriego, and J. A. Maestro, "Error detection technique for a median filter", *IEEE Transactions on Nuclear Science*, Vol. 64, No. 8, pp. 2219-2226, 2017.

[14] V. Kumar, A. Asati, and A. Gupta, "Low-latency median filter core for hardware implementation

of 5× 5 median filtering", *IET Image Processing*, Vol. 1, No. 1, pp. 927-934, 2017.
[15] W. T. Chen, P. Y. Chen, Y. C. Hsiao, and S. H. Lin, "A Low-Cost Design of 2D Median Filter", *IEEE Access*, Vol. 7, pp. 150623-150629, 2019.
[16] Sharanabasappa and P. Ravi Babu, "FPGA implementation of1D Median Filter architecture for de-noising application", *International Journal of Advanced Science and Technology*, Vol. 28, No. 9, pp.254-270, 2019.