International Journal of Electronics Engineering (ISSN: 0973-7383)
Volume 11 • Issue 1 pp. 584-593 Jan 2019-June 2019 www.csjournals.com

Novel Architecture for Active Noise Cancellation System Using Least Mean Square Algorithm

Saurabh R. Prasad[1*], Bhalchandra B. Godbole[2]
[1]Department of Elect. & Telecom. Engg, DKTE Society's Textile and Engg Institute, Ichalkaranji, India
[2]Department of Electronics Engineering, Karmaveer Bhaurao Patil College of Engineering, Satara, India

Abstract: Active noise control (ANC) is a promising area of research in which a processor estimates the noise contents of noisy signal and synthesizes noise of opposite polarity. When this opposite polarity noise is added to the noisy signal, the noise gets cancelled out and pure signal is obtained. Because of the dynamics of the noisy signal this ANC system requires the use of adaptive filter instead of conventional fixed filter for noise estimation. The real time hardware implementation of this system is challenging task because of complexity of algorithm and limitations of Field Programmable Gate Array (FPGA) board. So, in this paper we have implemented Least Mean Square (LMS) algorithm with Q_{18} notation instead of floating point architecture using on Spartan-6 FPGA board to reduce the hardware complexity. The result shows the comparison between floating point architecture and Q_{18} notation architecture and indicates the superiority of the later.
Keywords: adaptive filter; active noise cancellation; ANC; least mean square; LMS; floating point architecture; Q_{18} notation; spartan-6; FPGA; computational complexity.

1 INTRODUCTION

This paper implements an ANC system on Spartan-6 FPGA board using improved architecture. The filter is designed using the LMS algorithm due to its computational simplicity, robust behavior when implemented in finite-precision scenario and well understood convergence behavior.

An adaptive filter is a filter that self adjusts its transfer function according to an optimizing algorithm. Adaptive algorithm allows the filter to learn the initial statistics of the input signal and to track them afterwards for any further changes. The error signal, which is the difference between primary signal and estimated output of adaptive filter, is used to tune filter coefficients in order to minimize the cost function. The most commonly employed cost function is the mean square of the error signal (MSE).

Although the adaptive filters can be implemented as linear or non linear system, mostly they are a linear system and are much popular compared to their counterpart. Gabor in 1954 put forth the idea of a nonlinear adaptive filter using a Volterra series but that couldn't become much popular. Later there was development of nonlinear algorithms based on neural network and fuzzy logic. Fig. 1 shows the general plan of adaptive filter system.

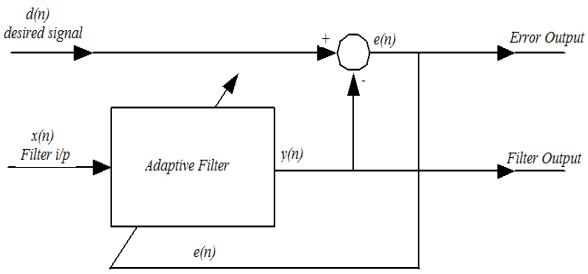

Fig. 1. Adaptive Filter Structure

The underlying digital filter in an adaptive filter is most commonly realized as FIR filter with transversal structure as shown in Fig. 2. This requires only delay line, adders, and multipliers as digital hardware resources. The adaptive filtering algorithm iteratively adjusts filter coefficients w_0, w_1... in order to minimize the cost function.

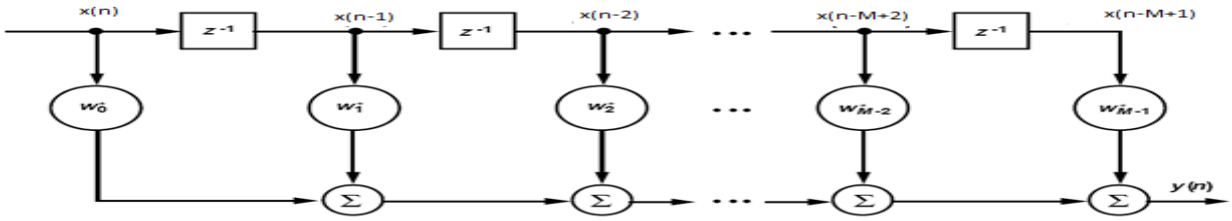

Fig. 2.Transversal FIR Filter

International Journal of Electronics Engineering (ISSN: 0973-7383)
Volume 11 • Issue 1 pp. 584-593 Jan 2019-June 2019 www.csjournals.com

1.1 Performance Parameters

Following are the performance parameters of the adaptive filters used o evaluate the effectiveness of the improved algorithm.

1.1.1 **Convergence rate**:

This is defined as the number of iterations required for the algorithm to converge to its steady state mean square error in case of stationary input. Thus a faster rate of convergence means that it requires less no of iterations to converge. Usually real time applications require faster rate of convergence.

1.1.2 Computational complexity:

The computational complexity of the algorithm influences the price of the processor needed to implement the adaptive filter. It can be evaluated with the number of operations the algorithm requires to accomplish a complete iteration. In addition, the size of memory for storing the data and program is also another important consideration. The need for reducing the computational complexity comes from two aspects. Firstly, for the inherently demanding algorithm like RLS, the O (L^2) complexity will occupy a large amount of computation and storage resources. Secondly, some applications like echo cancellation themselves are very much hardware demanding since in acoustic echo cancellation (AEC) problem, the echo paths spread over a relatively long length in time. Even for a sampling rate of 8 kHz, this would mean 800-2000 taps. In such situations, even the O (L) algorithm like NLMS is also much hardware demanding.

1.1.3 Numerical Robustness:

The implementation of adaptive filtering algorithms on a digital computer that has finite word-lengths, results in quantization errors. These errors sometimes can cause numerical instability of the adaptation algorithm. Adaptive filtering algorithm is numerically robust when its digital implementation using finite-word-length operations is stable.

1.1.4 Mathematical tractability:

An algorithm should be mathematically tractable, means that the convergence of the algorithm can be mathematically proved. Mathematical tractability is a desired feature of any adaptive algorithm.

1.1.5 Misadjustment:

As per the theory of optimal filter, no other filter than the optimal filter gives minimum MSE. Misadjustment is an important parameter in adaptive filtering which indicates how much closer a filter under consideration is to the optimal filter. The misadjustment is defined as the ratio of excess MSE to minimum MSE and represented as,

$$M = \frac{J_{ex}}{J_{\min}} \tag{8}$$

$$M = \frac{Jss - J\min}{J\min} \tag{9}$$

1.1.6 Stability

An algorithm is said to be stable if the mean-squared error converges to a finite value.

1.2 Applications of Adaptive Filters

There are following four categories of applications of adaptive filter.

1.2.1 **System Identification (Channel modeling)**

As shown in Fig. 3, system identification application requires both the unknown system and the adaptive filter driven by same input. After performing several iterations the filter estimates a linear model that closely matches with the unknown system.

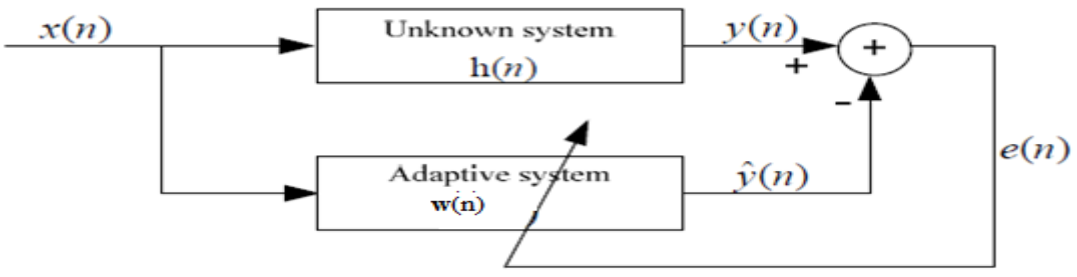

Fig.3. System Identification

1.2.2 Equalization (Inverse Channel Modeling)

As shown in Fig. 4, in equalization application unknown system and adaptive filter are connected in cascade and the input signal drives this cascade connection and delay line. To minimize the error signal, the adaptive algorithm brings the transfer function of the adaptive filter closer to the inverse of the unknown system's transfer function. The adaptive equalizers are widely used in routers and repeaters.

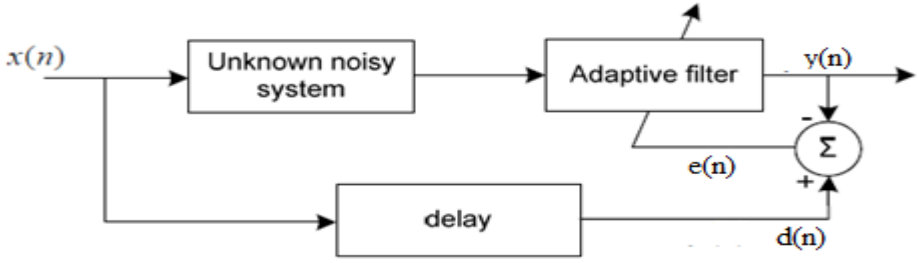

Fig.4. Equalization (Inverse Channel Modeling)

1.2.3 Signal Enhancement (Noise cancellation)

As shown in Fig. 5, in signal enhancement application, correlated noise is the input signal which drives the adaptive filter. The primary signal consists of a signal s(n) corrupted by an additive noise $n_1(n)$ and the filter input signal is x(n) which is correlated with $n_1(n)$ but uncorrelated with s(n). The adaptive filter output is estimate of noise $n_1(n)$ present in primary signal which gets subtracted to produce clean sound. This application can be found in hearing aids and noise cancellation system in aircraft, and outdoor news recording.

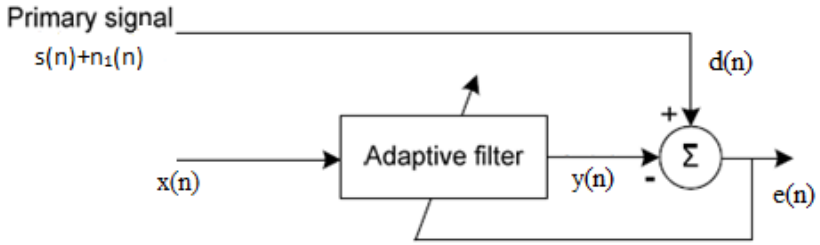

Fig.5. Signal Enhancement (Noise Cancellation)

Signal Enhancement application- Biomedical Signal Processing:

When signal and noise spectrum are distinct from each other and the characteristics of the signal, noise, and channel are known, fixed filters are the choice. But when there is a spectral overlap between signal and noise or when the characteristics of the input signal, noise, and dynamics of channel are unknown or change with time, fixed filters are of no use but adaptive filters only come for rescue. The practical example is as shown in Fig. 6 where ECG consists of fetal ECG as signal and mother's ECG as well as myographic signals as noise and both have the same frequency spectrum. In such situation, usually the noise is even stronger than the intended signal. These examples need adaptive filters.

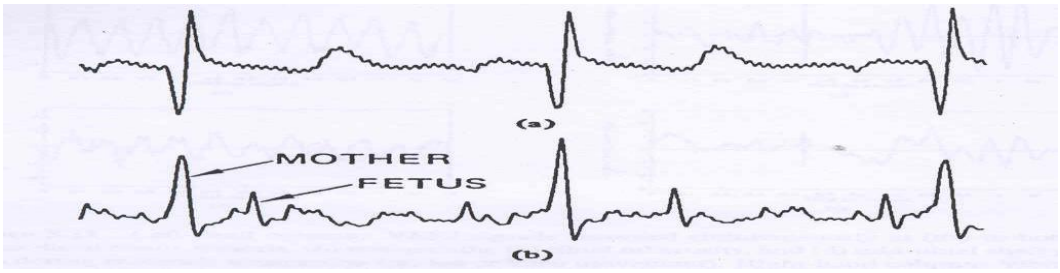

Fig.6. a) Clean ECG, b) ECG of fetus

1.2.4 Prediction (Beamforming)

If adaptive filter is configured as shown in Fig.7, it forms prediction system which is able to predict the present sample of the input signal using past values of the signal. This configuration is widely used in beamforming in smart antenna, where antenna can sense Direction of Arrival (DOA) of signal for proper beam steering. Another application of this system could be in speech coding for prediction based compression.

International Journal of Electronics Engineering (ISSN: 0973-7383)
Volume 11 • Issue 1 pp. 584-593 Jan 2019-June 2019 www.csjournals.com

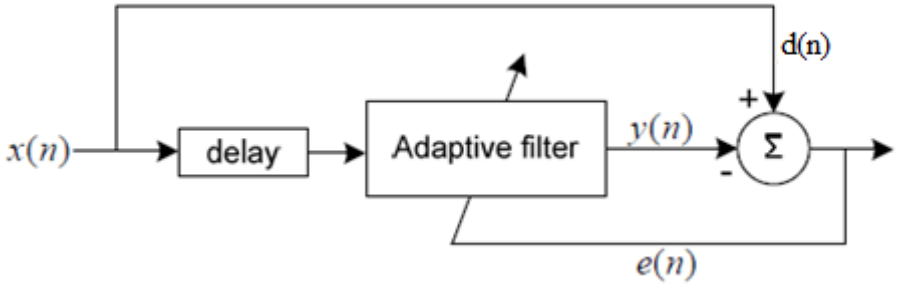

Fig.7. Prediction (Beamforming)

2.1 Wiener Filter

Wiener filter, which is also called as optimal filter is the basis of adaptive filter theory. Consider following notations where bold faced letters indicate vector and others indicate scalar quantities.

y(n) : estimated output of adaptive filter at current iteration
e(n) : output error signal at current iteration
d(n): desired signal (primary signal) at current iteration
J(w): Cost function to be minimized
$\mathbf{w}_{opt}$:optimal filter weight
x(n) : filter input (reference signal) at current iteration
R : Autocorrelation matrix of reference signal
p : cross correlation vector between input signal and error signal
w(n+1) : filter coefficient for next iteration
w(n) : filter coefficient at current iteration
μ : step size parameter

The output of adaptive filter obtained by convolution operation between filter weight and input signal is represented in (1). The instantaneous error signal which is the difference between desired signal and filter estimated output is represented in (2)

$$y(n) = \mathbf{w}^T \mathbf{x} \tag{1}$$

$$e(n) = d(n) - y(n) \tag{2}$$

LMS family of adaptive filtering uses Mean Square Error (MSE) as a Cost function which is represented in (3)

$$J(\mathbf{w}) = E\{e^2(n)\} \tag{3}$$

The optimization problem carried out to minimize the cost function solved by Wiener is represented in (4).

$$\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{p} \tag{4}$$

The optimal solution is a quadratic function of filter coefficients, and therefore possesses unique local minima as shown in Fig. 8 by bottom of the bowl shaped MSE surface plotted w.r.t first order filter coefficients w_0 and w_1.

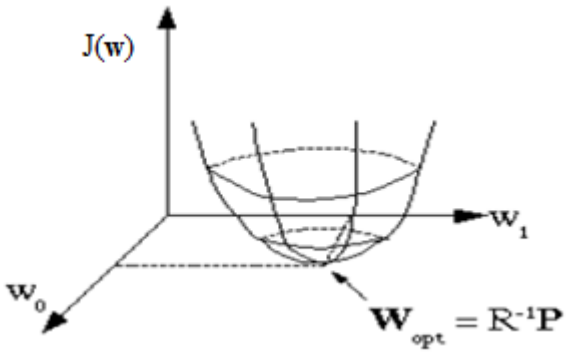

Fig. 8.Wiener solution for first order filter

Although Wiener gives optimal solution still it is of little practical significance because
1) It is computationally intensive as matrix inversion is required for its computation
2) It requires priori statistical information of the input signal in the form of autocorrelation matrix and cross correlation vector, which is generally not available.

So instead of using Wiener filtering solution directly as given in equation, the adaptive filters work in two phases; filtering phase and filter weight update phase to employ an iterative approach to find approximate optimal solution. LMS algorithm is the most popular of such iterative optimization algorithm.

2.2 LMS Algorithm:

The LMS algorithm has become immensely popular because performance, simplicity and stability of LMS algorithms outweigh other algorithms. The legendry LMS Algorithm was invented in 1959 by Bernard Widrow, and his first doctoral research scholar, Ted Hoff of Stanford University through their studies of pattern recognition. LMS algorithm stands as the benchmark against which all other adaptive filtering algorithms are judged. There are various variants of the LMS algorithm, and the original one mentioned in this section is called as standard LMS algorithm. In stationary environment, the LMS filter is expected to converge to the Wiener filter, and in nonstationary environments, the filter is expected to track time variations of the signal and vary the coefficients accordingly. Equation (5) illustrates the weight update equation of LMS algorithm.

$\mathbf{w}(n+1)=\mathbf{w}(n)+\mu\mathbf{x}(n)e(n)$ (5)

Computational complexity of LMS: This algorithm requires N+1 multiplication and N additions for a filter of N taps for coefficient update. The filtering process requires N multiplications and N-1 additions. The generation of error equation requires one addition. Therefore total number of operations per iteration is 2N+1 multiplication and 2N addition and consequently this algorithm is called as O (N) algorithm.

LMS Algorithm

Initialization:

$\mathbf{w}(0)=\mathbf{0}$

Algorithm:

For n=0: iterations

$y(n)= \mathbf{w}^T(n)\mathbf{x}(n)$

$e(n)= d(n)-y(n)$

$\mathbf{w}(n+1)=\mathbf{w}(n)+\mu\,\mathbf{x}(n)\,e(n)$

end

2.3 Stability Condition for LMS Algorithm

Stability and causality are the two basic requirements that must be fulfilled by any digital filter. This is because a non causal filter would be non-realizable and an unstable filter would be of no use. The LMS algorithm is converges in mean square if and only if the stepsize parameter μ satisfy (6) where λ_{max} is largest eigenvalue of the autocorrelation matrix of input signal. But in practical situations, this equation to calculate stepsize is not much useful because the eigenvalues of the autocorrelation matrix are not known, and the more useful equation for the same is given in (7) where $\|x(n)\|$ represents Euclidean norm of the input signal vector whose squared term represents the power of the signal which is usually known or can be estimated a priori [3].

$$0 < \mu < \frac{2}{\lambda_{max}} \quad (6)$$

$$0 < \mu < \frac{2}{\|x(n)\|^2} \quad (7)$$

3 Proposed Efficient Architecture for LMS

The proposed real time audio de-noising is as shown in Fig.9. The input noise-free audio signal is converted into digital format by passing it through ADC present in the input audio driver of Xilinx ATLYS FPGA board. A pseudo-random noise generator circuit is used to generate noise within FPGA which is then added with the audio signal to produce noisy audio signal. This noisy signal is then fed to audio codec (ac97) to convert it into frames of 64 audio samples and finally applied to modified LMS algorithm block to obtain the filtered output. The filtering algorithm is dynamic; as a result it will take some samples to calculate the factors for de-noising. This de-noised signal is again converted to analog through DAC available in the audio codec (ac97).

International Journal of Electronics Engineering (ISSN: 0973-7383)
Volume 11 • Issue 1 pp. 584-593 Jan 2019-June 2019 www.csjournals.com

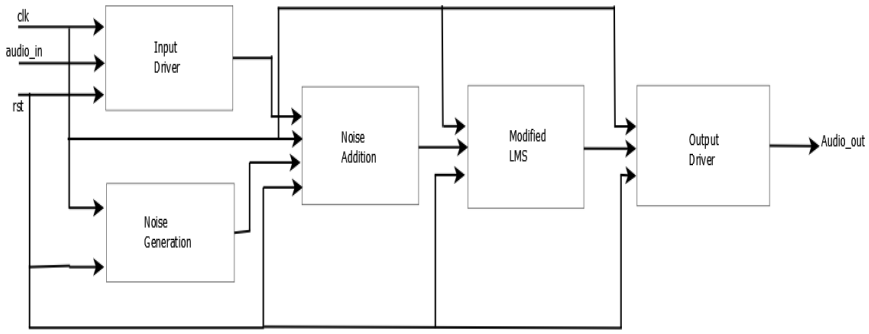

Fig.9. Block Diagram of Active Noise Cancellation Using FPGA

3.1 Audio Driver:

The driver for audio signal has to be configured at both input and output side to interface with FPGA. The input driver consists of standard ac'97codec. The register value of the standard codec has to be set with some value to recognize the samples. In our work we set the register through FSM model using data sheet. The inputs to the controller embrace the most FPGA generator, a lively low reset, a serial information in line, a 12.288 MHz bit clock from the ac'97 chip, a three bit supply selector (slide switches) and a five bit volume management (slide switches). The output to the controller embrace a adjust signal, serial information output, an ac97 active low reset signal for initializing the ac97. There exists a control pin to adjust the most ac'97 controller with the command state machine. The single frame of audio signal is of 20µs and at every 20 µs of data is transmitted.

3.2 Noise Generation and noise addition:

To generate noise we used normal Pseudo Random Noise Generator (PRNG) circuits whose output is then added to the audio signal to generate noisy signal.

3.3 Modified Architecture for LMS Algorithm:

The noisy audio signal is de-noised by adaptive filter implemented with modified LMS architecture. In most of the cases, noise present in a noisy signal doesn't remain stationary but changes with time, and also the noise sources are unknown. So, in such situation fixed filter will be of no use, but an adaptive filter which changes itself according to changing noise statistics can perform suitably. Therefore, we have designed an adaptive noise removal filter based on LMS algorithm. As shown in Fig. 10 the desired signal d_k as represented in (8) is formed by adding signal (S_k) and noise (N_k).

$$d_k = S_k + N_k \quad (8)$$

The input signal (reference signal) to filter represented in (9) is generated when (8) is delayed by one sample to de-correlate the noise content in input to adaptive filter from the signal in d_k.

$$x_k = (S_{kd} + N_{kd}) \quad (9)$$

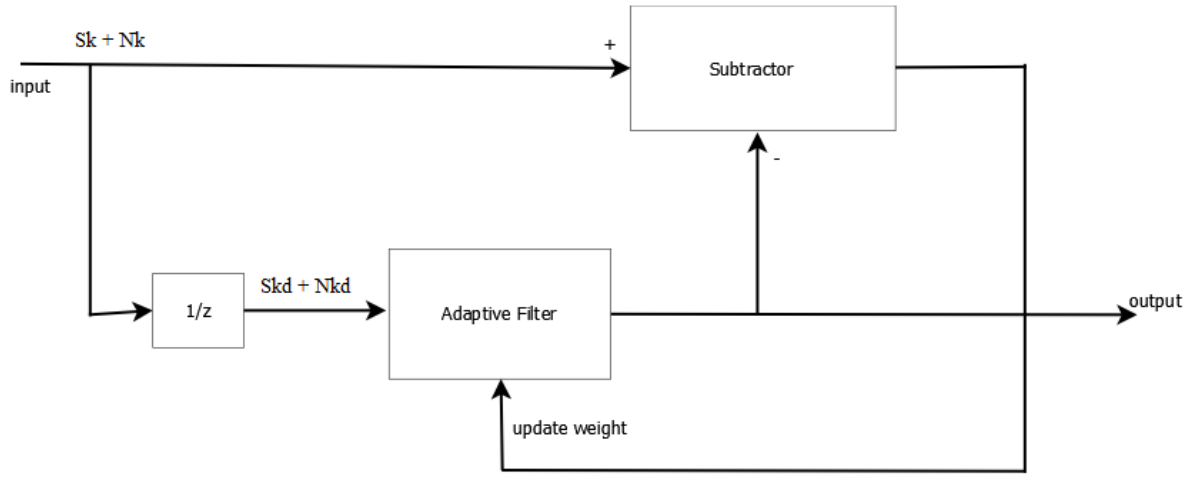

Fig.10. General Schematics of Active Noise Cancellation

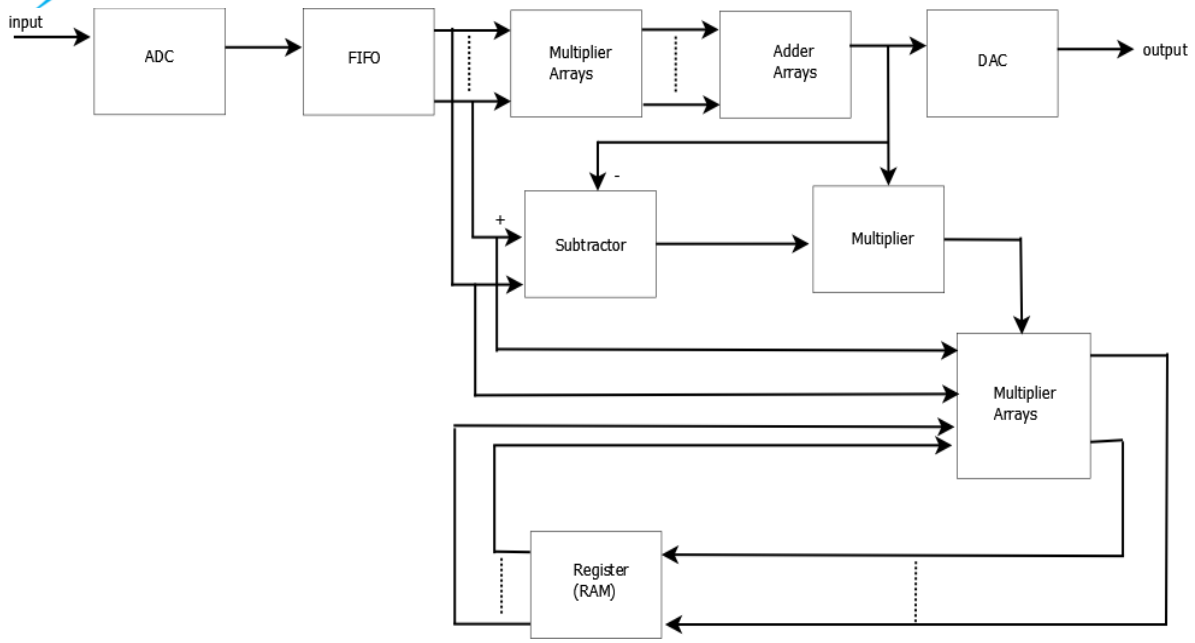

Fig.11. Architecture of FPGA for ANC System

Fig.11 represents the implemented architecture of the LMS algorithm. Basically the architecture has been designed with FIFO register (RAM) which in turn gives the optimized result. The multiplier blocks are optimized by using shifters wherever constant filter coefficients exists. This helps in improving the speed of the entire system. It adjusts the filter weight values at each sampling period by calculating y_k using (10)

$$y_k = \sum_{i=0}^{N-1} w_k(i) x_{k-i} \qquad (10)$$

This signal y_k is then used to calculate the error as per (2). The filter coefficients are updated by using (5). To reduce the MSE up to the maximum limit we must have to make the gradient value as zero which is not possible in practical case. In reality we can make this nearer to zero by fixing small stepsize (μ). In our implementation the value of μ is predicted from (7) to ensure filter doesn't diverge. The calculation of μ value is implemented by a simple barrel shifters followed by a FIFO. It is important to keep μ values nearer to zero at initialization for proper operation.

As per adaptive filtering theory, an underestimated adaptive filter severely degrades the performance while an overestimated filter doesn't. Therefore, the order of the adaptive filter in our experimental work is chosen to be 29.

The Hardware utilization of the implemented algorithm is shown in Table 1 which fits in the available resources of Spartan-6 FPGA board. Therefore, for this reason, and also because of ready availability and expertise, the algorithm is implemented using this board. The entire proposed model uses 1737 slice registers, 2181 slice LUTs with maximum operating speed of 110MHz. It is observed that the total system speed is reduced than LMS Audio Driver block, since the wire routing between blocks increases delay and decreases the speed in turn.

Table 1: Hardware Utilization:

Parameters	LMS Block	Audio Driver	Total Model
Slice Registers (CLB)	1439	163	1737
Slice LUTs	1860	149	2181
LUT-FF Pairs	605	104	729
DSP48A1s	48	0	48
Frequency	112.712	123.264	110.807

The comparative hardware utilization of different LMS algorithms by various implementations is presented in Table 2. In existing techniques [2, 3] all the coefficient values are subjected for direct multiplication for which a dedicated multipliers are used. Further, floating point computations are used to deal with real data. But, in our work, we have used shifter for constant multiplication and Q_{18} notation instead of floating point architecture

which helps in reducing the hardware resources. Q_{18} notation is normally used to implement fractional number in binary format. In this format total 19 bit is considered. In those 19 bits first MSB bit represent sign of the value and the remaining bits are used to generate fractional number.

Eg: +0.25 = 0… 010……. 00

Sign 18 bit

The ac97' codec uses needs notation to configure itself. So we have converted the values given in the datasheet of ac97' into this notation format for proper configuring.

Table2. Comparisons with Existing Technique

Parameters	Aishwaria and Vijayabhaskar [1]	Sang Park [2]	Nilesh S. Satpute et al., [3]	Jebin and Ramya [4]	Proposed
Slice Registers	2098	2743	2190	----	1421
Fully used FF Pairs	2011	2743	2190	----	605
Maximum Frequency (MHz)	----	----	----	105	112.712

4 Results

The implementation of the LMS algorithm with modified architecture is evaluated for its satisfactory performance with respect to PSNR graph.

4.1 RTL Schematic of Top Module:

HDL (Hardware description Language) is the computer language used to describe and program a digital circuit like FPGA and when it is synthesizable then it is considered RTL (Register-transfer level). The popular HDL programming languages are Verilog and VHDL and the later is used here for programming of FPGA. There are three levels of abstraction; namely Behavioral, RTL, and Gate-level. RTL is a level of abstraction that we usually deal with. The three levels of abstraction are explained below:

Behavioral has the highest layer of abstraction which describes the overall behavior and is often non-synthesizable, but is useful for verification.

RTL describes the hardware by implying logic, defining flip-flops, latches and how data is transferred between them. This is synthesizable; synthesis may alter/optimize the logic used but not behavior.

Gate level is a design using the base logic gates (NAND, NOR, AND, OR, MUX, FLIP-FLOP). It does not need to be synthesized. This has the lowest level of abstraction. It is the logic gates that will be used on the chip, but it lacks positional information.

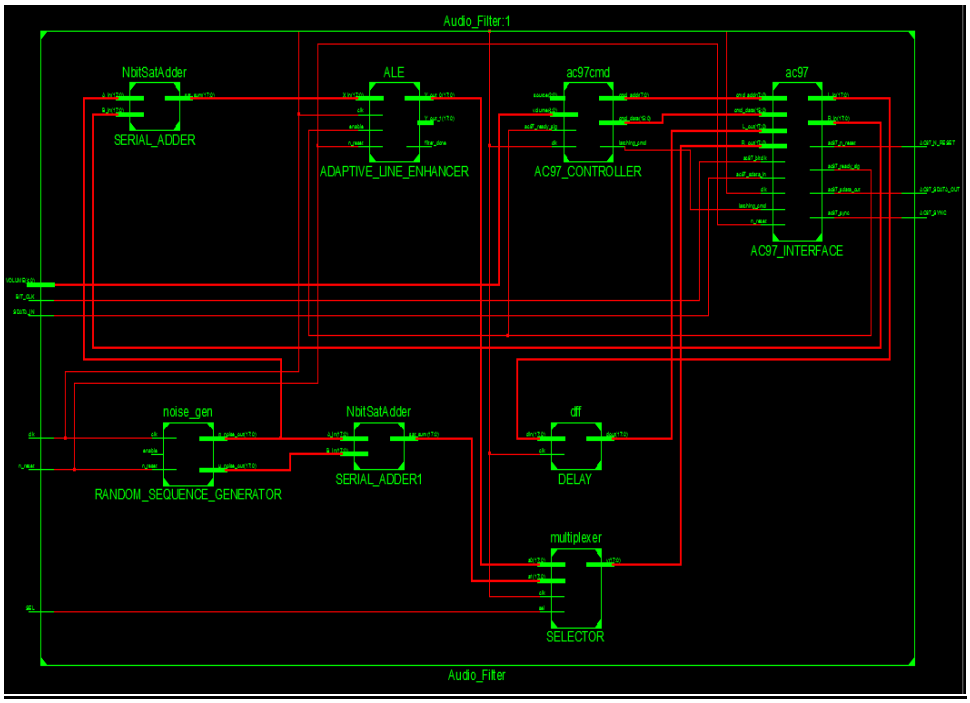

Fig.12. RTL schematic after synthesizing on Xilinx tool

International Journal of Electronics Engineering (ISSN: 0973-7383)
Volume 11 • Issue 1 pp. 584-593 Jan 2019-June 2019 www.csjournals.com

4.2 Modified LMS

The schematics of modified LMS are shown in Fig.13.

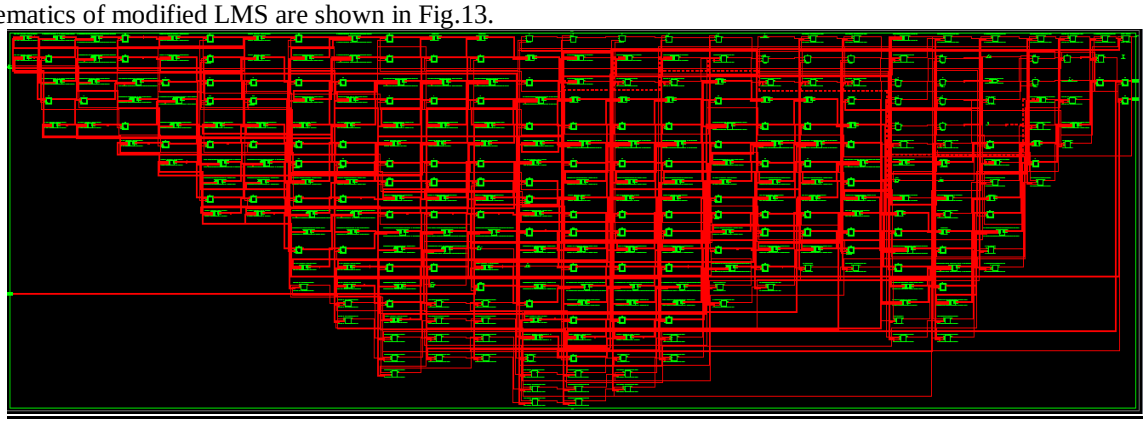

Fig.13. Schematics of Modified LMS

4.3 Convergence of data output with input

The diagram for convergence of data output and input is as shown in Fig. 14.

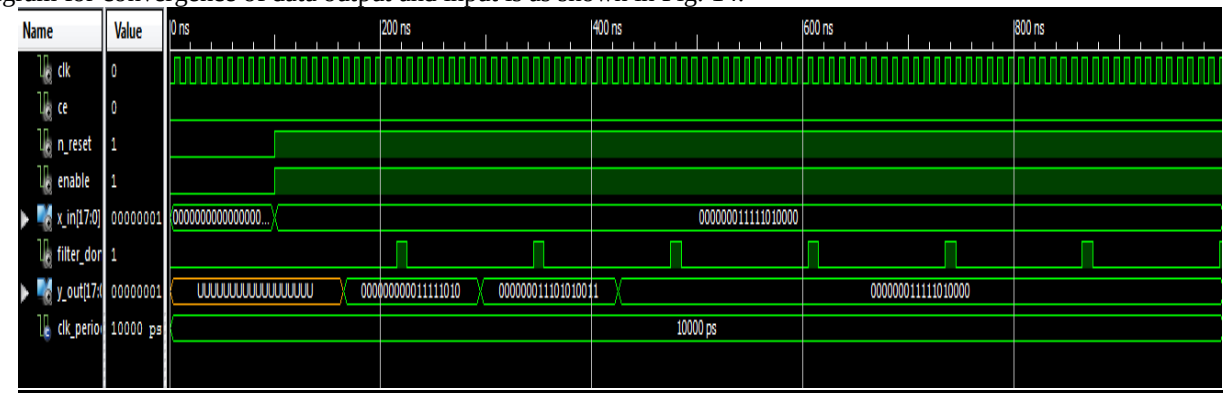

Fig.14. Diagram showing convergence of data o/p with i/p

4.4 Audio Driver

RTL diagram of audio driver is as shown in Fig.15.

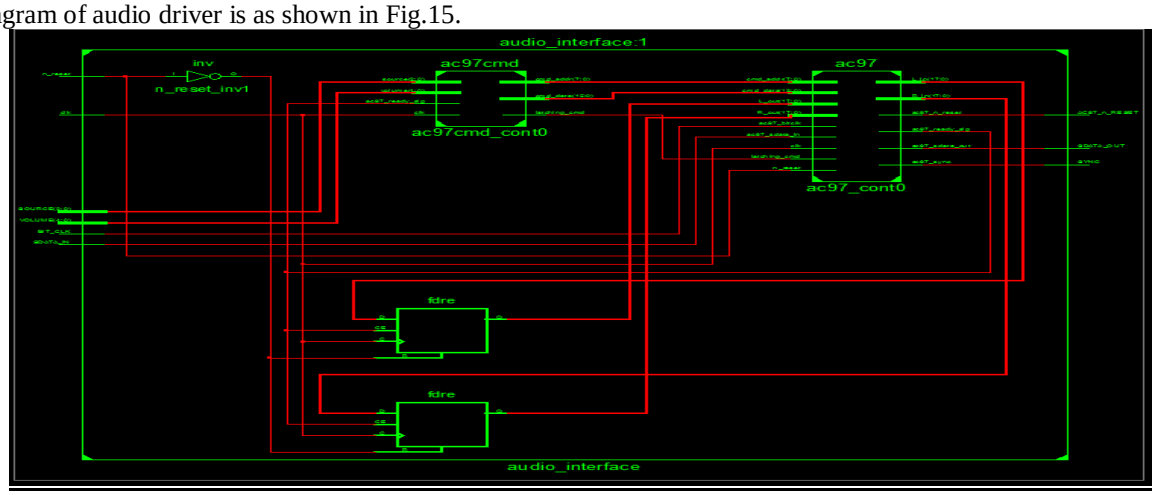

Fig.15. RTL Diagram of Audio Driver Implementation

4.4 FPGA Placement

At the FPGA placement stage, slot assignment problem is solved by taking netlist as input. Next is the routing in which various types of logic blocks need to be interconnected. The result of placement is a mapping of all blocks onto physical resources of target FPGA without violation of all constraints. A very common constraint is minimizing the total wire length among the route signals, because minimum wire length reduces the values of signal delay and power. The FPGA placement diagram of implemented adaptive ANC system is shown in Fig.16.

International Journal of Electronics Engineering (ISSN: 0973-7383)
Volume 11 • Issue 1 pp. 584-593 Jan 2019-June 2019 www.csjournals.com

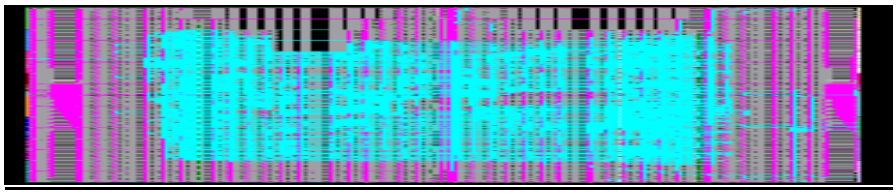

Fig.16. FPGA Placement Diagram

4.5 PSNR Graph

The PSNR graph plotted in Fig.17 shows that the PSNR values are varying. Since it takes 65536 samples at a time inside the system generator block so till that time redundant PSNR is obtained and only after every 65536 samples an accurate PSNR is obtained.

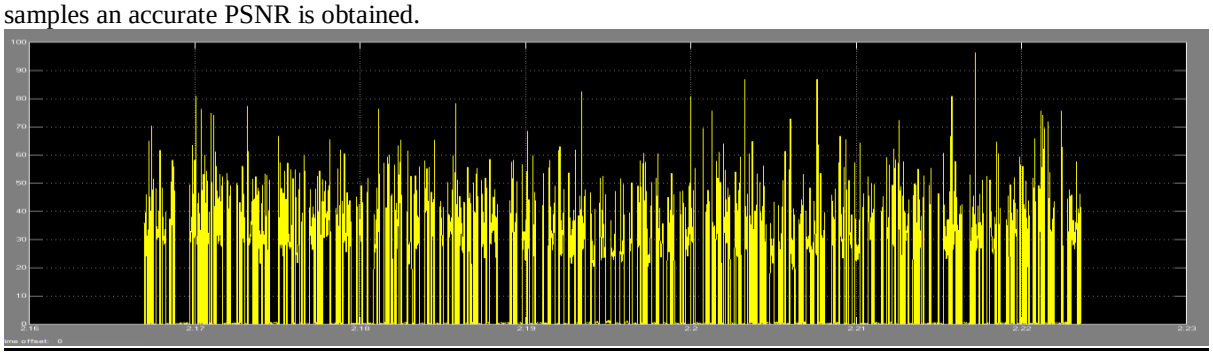

Fig.17. PSNR graph

Conclusion

An ANC system is successfully implemented on Spartan-6 FPGA board using Q18 notation instead of floating point architecture which substantially reduces FPGA device utilization. The results with modified low complexity ANC system are similar to conventional high complexity method.

References:

[1] Aishwaria C and Vijayabhaskar R, "Enhanced Pipelined Architecture for Adaptive FIR Filter based on Distributed Arithmetic", International Journal of Advanced Information Science and Technology, Vol. 23, No. 23, 2014.

[2] Sang Yoon Park, June 2013, "Low power, High Throughput, and Low-Area Adaptive FIR Filter Based on Distributed Arithmetic", IEEE Transactions on circuits and systems-II: Express Briefs,Vol.60, Issue. 6, pp. 346-350, 2013.

[3] Nilesh S. Satpute, Sanjay B. Tembhurne and Vipin S. Bhure, "LMS Algorithm and Distributed Arithmetic Based Adaptive FIR Filter with Low Area Complexity", International Journal of Advanced Research in Computer and Communication Engineering, Vol. 6, Issue. 6, pp. 355-359, 2015.

[4] Jebin Roy and Ramya, "Low Power and Low Area Adaptive FIR Filter based on Arithmetic Algorithm" International Journal of Scientific and Research Publications, Volume 4, Issue 3, 2014.

[5] M. Lapointe, P. Fortier, and H. T. Huynh, "Fast parallel realization of the LMS algorithm in O(logN) computation time", 15th Biennial Symposium on Communications,

[6] Kingston, Canada, June 1990.N. Weste and K. Eshraghian, "Principles of CMOS VLSI design: A system perspective", Addison-Wesley, Reading, 1985.

[7] A. Avizienis, "Signed-digit number representations for fast Parallel arithmetic", IRE Trans. Electron. Comput., vol. 10, pp. 389-400, September, 1961.