

# Design and implementation of high-speed adaptive filter for speech enhancement

B. Mysura Reddy<sup>1</sup>, Sk. Rijwana<sup>2</sup>, T. Sai Dinesh<sup>3</sup>

<sup>1</sup>Assistant Professor, <sup>2,3</sup>Student

Electronics and Communication Engineering

N.B.K.R Institute of Science and Technology, Andhra Pradesh, India

**Abstract: - The FPGA implementation of retimed high speed adaptive filter architectures for voice enhancement is shown in this research. Many high-speed adaptive filtering techniques for noise cancellation are implemented in this work. The clock speed, hardware requirements, delay, and cost of various VLSI implementations have been seen to vary significantly. The VLSI implementation and performance analysis provide critical information about the structure of an algorithm, such as hardware requirements, power consumption, and real-time performance. The implemented structures' performance was evaluated in terms of operating frequency, maximum combinational path delay, latency, and power consumption.**

**Keywords: - Adaptive filter, Xilinx, DLMS, High speed.**

## I. INTRODUCTION

Noise levels have increased dramatically during the previous decade, particularly in urban areas. Noisy settings are increasingly an inescapable feature of city life. Rapid industrialization, exposure to various electronic gadgets such as TV, music systems, mobile phones, and household electrical equipment such as air conditioners, exhaust fans, mixer grinders, refrigerators, and washing machines are major drivers of ever-increasing noise levels. Many of these noises produced by various noise sources have an impact on the quality of communication between people when they are exposed to noisy situations.

Several types of noise damage the desired voice signal when it is transported from one point to another. The noise could be background noise at the source or channel noise. These noises are caused by time-varying physical processes, the majority of which are unknown and unpredictable in advance. Because most modifications are unknown, only adaptive filters can track and reduce noise from the target signal (Haykin 2008). Adaptive noise cancellation is a significant use of adaptive filters, which have been employed in a variety of digital signal processing systems. Noise reduction from ECG signals, noise reduction from speech signals, and so forth. 2011 (Bahoura and Ezzaidi).

Noise reduction from voice signals is a critical field of research. Several adaptive algorithms have been developed and improved upon in the past by various researchers in order to reduce unwanted noise levels. Hardware implementation of voice enhancement techniques has recently emerged as a significant topic of study. Several hardware platforms have emerged as a result of the rapid advancement of integrated circuit technology over the last decade or two.

These hardware platforms allow designers an option to implement a prototype hardware for an algorithm in their research lab themselves in a short period of time. Prototyping is required because it provides vital information about the structure of an algorithm, such as hardware requirements, power consumption, structural complexity, and real-time performance (operating frequency, critical path delay and latency). Before tape out, the performance of the algorithm structure produced through prototyping is regarded as an important barometer. Prototyping has substantially reduced costs and a shorter time to market.

FPGAs have emerged as an important platform for signal processing applications due to their highly parallel topologies and huge number of arithmetic circuits such as adders and registers that allow high levels of pipelining to be successfully used. FPGAs provide the additional benefits of high processing capability, inherent stability, and low power consumption (Yi and Woods 2005; Elhossini et al. 2006; Mustafa et al. 2009; Fohl and Matthies 2009; Mohanty et al. 2017).

Many adaptive filtering structures are designed in this paper. The clock speed, hardware requirements, delay, and cost of various VLSI implementations have been seen to vary significantly. It has also been discovered that retimed adaptive filter architectures outperform ordinary un-retimed structures.

## II. ADAPTIVE FILTER

An adaptive filter is one which adapts its coefficients according to changes in its surroundings. The adaptive filter is self-adjusting and tracking (Haykin 2008). It can be implemented as an IIR (infinite impulse response) or FIR (finite impulse response) filter. However, because IIR filters are recursive, they become unstable when their pole moves outside of the unit circle in the Z-plane. Figure 1 depicts a simplified block diagram of an adaptive filter.

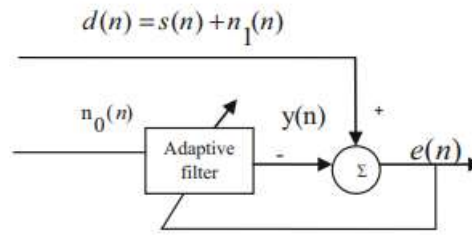


Fig. 1 LMS adaptive filter algorithm

The circuit in Figure 1 has two inputs: primary and reference. The initial signal  $s$  is corrupted by noise  $n_1(n)$ . The reference input is noise  $n_0(n)$ .  $n_0(n)$  is unrelated to  $s(n)$ , yet it is related to  $n_1(n)$  in an unknown way. Both  $n_0(n)$  and  $n_1(n)$  are uncorrelated with  $s(n)$ , implying the following:

$$d(n) = s(n) + n_1(n) \quad \text{for all } n \tag{1}$$

$$E[s(n)n_1(n)] = 0 \quad \text{for all } n \tag{2}$$

$$E[s(n)n_0(n)] = 0 \quad \text{for all } n \tag{3}$$

$$E[n_1(n)n_0(n)] = p(n) \quad \text{for all } n \tag{4}$$

where  $d(n)$  represents the desired signal,  $E[*]$  represents the expectation, and  $p(n)$  represents an unknown correlation between  $n_1(n)$  and  $n_0(n)$ .

In the best case, the error signal  $e(n)$  should contain the original signal  $s(n)$  (Elhossini et al. 2006).

Because of its simplicity, LMS is the most often used algorithm in adaptive filtering. A gradient descent algorithm is used. Adaptive filter taps are updated by an amount proportional to the error surface gradient's instantaneous estimate (Haykin 2008). The following equations are used to apply the LMS algorithm to the noise cancelling problem:

$$u(n) = s(n) + n_1(n) \tag{5}$$

$$d(n) = u(n) \tag{6}$$

$$y(n) = w(n)^T u'(n) \tag{7}$$

$$e(n) = d(n) - y(n) \tag{8}$$

$$w(n+1) = w(n) + 2\mu e(n)u'(n) \tag{9}$$

where  $u(n)$  is a signal made up of  $s(n)$  and  $n_1(n)$ ;  $d(n)$  is the intended signal;  $y(n)$  is the ANC output;  $w(n)$  is the tap weight vector;  $u_0(n)$  is a vector made up of  $n_0(n)$  with the same size as  $w(n)$ ;  $e(n)$  is the error signal; and  $\mu$  is the step-size parameter (Dhal et al. 2015; Kar et al. 2014).

In several practical applications, such as decision directed adaptive equalisation, adaptive reference echo cancellation, development of adaptive algorithms employing parallel architectures etc., the LMS adaptation scheme puts a critical limit on its implementation (Long et al. 1989).

Furthermore, the original LMS algorithm structures have a long critical path latency and a slow convergence rate. By changing the original structure with various retiming approaches, the critical path delay can be minimised. Nevertheless, in order to use these retiming strategies, the LMS algorithm must be programmed with delayed coefficients. Delayed LMS is the name given to this modified type of LMS (DLMS). The DLMS method can be obtained by inserting a delay in the LMS algorithm's error feedback loop, as shown in Fig. 2.

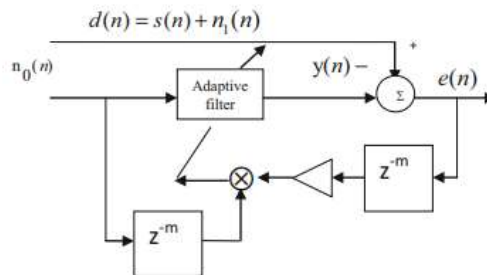


Fig. 2 Delayed LMS adaptive filter algorithm

The DLMS algorithm can be expressed by the following equations:

$$e(n-m) = d(n-m) - y(n-m) \tag{10}$$

$$w(n+1) = w(n) + 2\mu e(n-m)u'(n-m) \tag{11}$$

Where  $m$  is the number of delay elements and  $y(n-m)$  is a delayed version of  $y(n)$

### III. RETIMING TECHNIQUE

In VLSI circuit design, retiming is a highly helpful approach. A circuit can be enhanced via retiming to have a short critical path delay, a high throughput rate, and a low power consumption. The positions of delay elements in a circuit are modified with this approach, i.e. the delays are redistributed without changing the circuit's input/output characteristics.

The retimed approach relocates delay components in the circuit to shorten its critical path. A data-flow graph can be used to describe retiming on a filter structure (DFG). Cutset retiming and pipelining are two particular examples of retiming. A cutset is a collection of edges that can be extracted from the original DFG to form two subgraphs. If we designate the two unconnected subgraphs G1 and G2, then cutset retiming consists of adding k delays to each edge from G1 to G2 and subtracting k delays from each edge from G2 to G1.

The critical route in the 4-tap DF-LMS construction shown in Fig. 3 consists of two multipliers and five adders, which restrict the circuit's maximum operating frequency. This critical route must be lowered to get the retimed DF-RDLMS structure by introducing delays between these combinational adders and multipliers. To do this, delays must first be incorporated into the filter structure, and then the positions of these delays must be modified so that the filter's input and output properties do not change. The delay elements in a DF-LMS filter structure are provided by  $3N - 2$ , where N is the filter length. As a result, the delay elements for a 4-tap filter construction are ten. This structure's weight update equation is given by

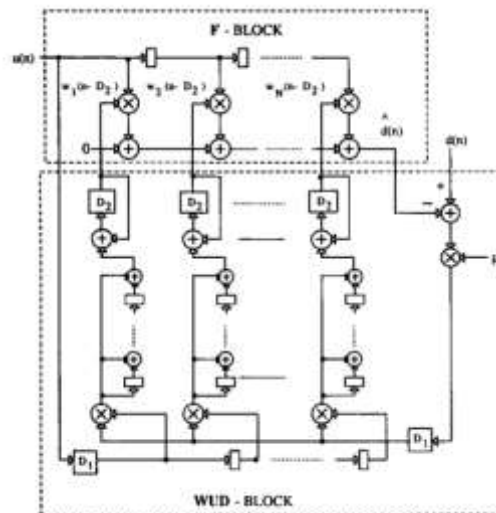
$$w(n + 1) = w(n) + 2\mu e(n)x'(n) \tag{12}$$

The Delayed LMS (DF-DLMS) structure will have  $5N$  (20 for 4-tap filter) delays. The value of m in the delayed LMS algorithm in Fig. 2 will be five for a 4-tap filter. As a result, five delays will be induced in the path of  $e(n)$  and  $u(n)$ . For Delayed LMS, the relevant weight update equation is provided by

$$w(n + 1) = w(n) + 2\mu e(n - m)x'(n - m) \tag{13}$$

The number of m is selected to be five because there are five horizontal cutsets in a 4-tap DF-LMS structure and one delay is required to compensate for each horizontal cutset. These delays must be redistributed in order to obtain the retimed DF-RDLMS structure.

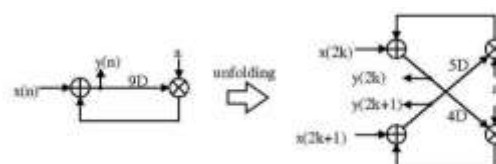
Pipelining is a technique was shown in figure 3 that offers increase in clock rate of the system and can be used to provide less power. This is achieved by placing latches in the feed forward cut set and the critical path is reduced. The sample rate is thus increased when pipelining is done.



**Fig. 3 Pipelined LMS adaptive filter algorithm**

Figure 4 shows a transformation technique called unfolding that is used to reduce the critical path by a factor of J. Loop unrolling a system with J delivers great speed but has a negative impact on area.

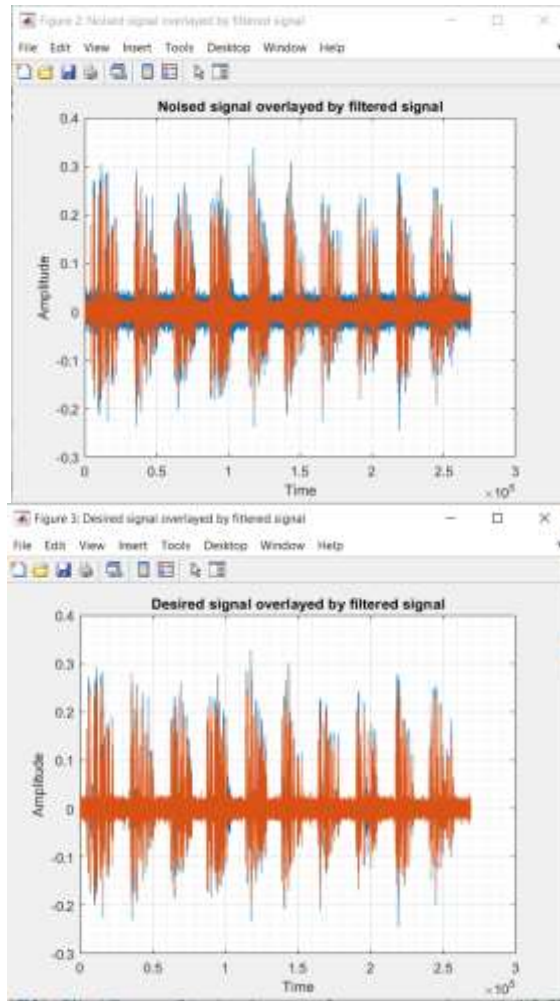
**Fig. 4 unfolded LMS adaptive filter algorithm**



**Fig. 5 unfolded technique**

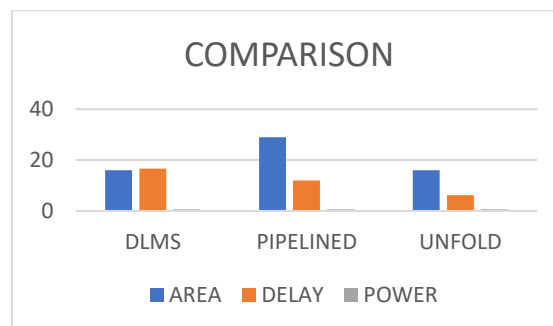
Figure 5 depicts a transformation strategy of duplicating the functional blocks to boost the throughput of the DSP programme while preserving its functional behaviour at its outputs.

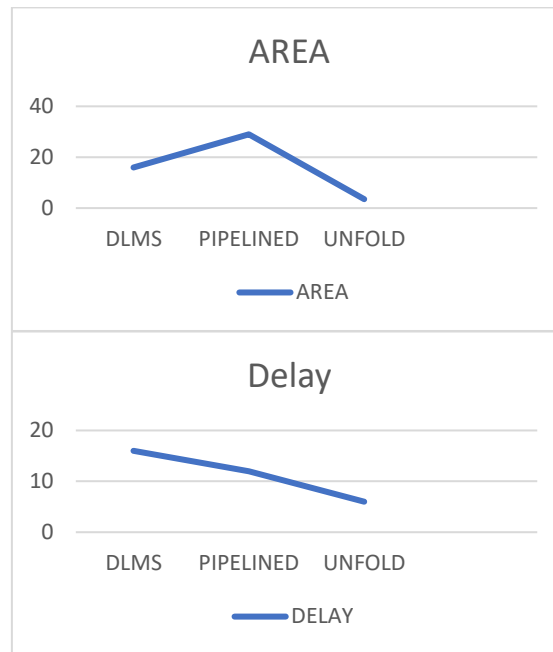
**IV. Results and discussion**



**Fig. 6 MATLAB Simulation of filtered signal**

**Table 1:- Comparison Table**





TECHNIQUE	AREA(LUT)	DELAY(ns)	POWER-W(STATIC)
DLMS-4TAP	16	16.610	0.060
PIPELINED-4TAP	29	12.026	0.060
UNFOLD-4TAP	16	6.216	0.060

**V. CONCLUSION**

This paper develops both traditional and high-speed adaptive filter designs. Figure 6 shows a MATLAB simulation of an adaptive LMS. Table 1 provided a table of adaptive filter comparisons. The adaptive filter was created using the DLMS, pipelined, and unfold techniques. As compared to other procedures, the unfolded technique produces superior outcomes.

**REFERENCES**

1. Bahoura M, Ezzaidi H (2010) FPGA-implementation of waveletbased denoising technique to remove power-line interference from ECG signal. In: 10th IEEE international conference on information technology and applications in biomedicine (ITAB), Corfu, Greece, pp 1–4
2. Bahoura M, Ezzaidi H (2011) FPGA implementation of parallel and sequential architectures for adaptive noise cancellation. In: Circuits, systems and signal processing, vol 30, pp 1521–1548. Springer, SP Birkhauser Verlag
3. Boston Dai J, Wang Y (2010) NLMS adaptive algorithm implement based on FPGA. In: Third international conference on intelligent networks and intelligent systems, IEEE, 2010, Shenyang, China, pp 422–425. <https://doi.org/10.1109/icinis.2010.97>
4. Dhal M, Ghosh M, Goel P, Kar A, Mohapatra S, Chandra M (2015) A unique adaptive noise canceller with advanced variable-step BLMS algorithm. In: 2015 international conference on advances in computing communications and informatics (ICACCI) Kochi, India, pp 178–183
5. Elhossini A, Areibi S, Dony R (2006) An FPGA implementation of the LMS adaptive filter for audio processing. In: IEEE International Conference on Reconfigurable Computing and FPGA's, 2006. ReConFig 2006. IEEE, San Luis Potosi, Mexico, pp 1–8
6. Fohl W, Matthies J (2009) A FPGA based adaptive noise cancelling system. In: Proceedings of the 12th international conference on digital audio effects (DAFX-09), Como, Italy, September 01–04
7. Kar A, Chanda AP, Mohapatra S, Chandra M (2014) An improved filtered-x least mean square algorithm for acoustic noise suppression. In: Advanced computing, networking and informatics-volume 1, smart innovation, systems and technologies, vol 27. Springer, Cham, pp 25–32
8. Long G, Ling F, Proakis JG (1989) The LMS algorithm with delayed coefficient adaptation. IEEE Trans Acoust Speech Signal Process 37(9):1397–1405
9. Mohanty BK, Singh G, Panda G (2017) Hardware design for VLSI implementation of FxLMS- and FsLMS-based active noise controllers. Circuits Syst Signal Process (Springer) 36(2):447–473 (first online April 2016)

10. Monteiro J, Devadas S, Ghosh A (1993) Retiming sequential circuits for low power. In: Proceedings of international conference on computer aided design (ICCAD-1993), Santa Clara, California, Nov 7–11, 1993, pp 398–402
11. Mustafa R, Umat C, Ali MAM, Al-asady AD (2009) Design and implementation of least mean square adaptive filter on altera cyclone II field programmable gate array for active noise control. In: IEEE Symposium on Industrial Electronics & Applications (ISIEA 2009), vol 1, Kuala Lumpur, Malaysia, pp 479–484
12. Parhi KK (2010) VLSI digital Signal processing systems: design and implementation, 1st edn. Wiley, India
13. Rizwan S (2008) Retimed decomposed serial Berlekamp–Massey (BM) architecture for high-speed Reed–Solomon decoding. In: 21st international conference on VLSI design (VLSID 2008), Hyderabad, India, pp 53–58
14. Samudravijaya K, Rao PVS, Agrawal SS (2000) Hindi speech database. In: Proceedings of international conference on spoken language processing, vol 4. ICSLP-2000, Beijing, China, pp 456–459
15. Shenming W, Suntiamorntut W, Jindapetch N, Qiufan J (2011) Scheduling and resources sharing technique for adaptive LMS filter. In: The 8th electrical engineering/electronics computer telecommunications and information technology (ECTI) association of Thailand, conference 2011, pp 114–117
16. Yagain D, Krishna AV, Chennapoor S (2012) Design optimization platform for synthesizable high speed digital filters using retiming technique. In: 2012 10th IEEE international conference on semiconductor electronics (ICSE), Kuala Lumpur, Malaysia, pp 551–555 Yi
17. Y, Woods R (2005) High speed FPGA-based implementations of delayed-LMS filters. *J VLSI Signal Process* 39:113–131