

# Novel approach of High Effective 64 Tap Fixed-Point DLMS Adaptive Filter

Lakshmi Sai Mounika Devi Vegesna<sup>#1</sup> and P. Srinivas<sup>\*2</sup>

<sup>#</sup>M. Tech (VLSI), Department of ECE, BVC College of Engineering, Rajahmundry, A.P, India Email.id:vls.mounika@gmail.com. \* Associate Professor, Department of ECE, BVC College of Engineering, Rajahmundry, A.P,India Email.id: srikireet@gmail.com

Abstract— Modern Field Programmable Gate Arrays (FPGAs) include the resources needed to design efficient filtering structures of delayed least mean square adaptive filter. From the architecture, the low adaption delay, Area and power are synthesized. Fixed point implementation scheme architecture with bit level clipping is enhanced Adaptive filters learn the statistics of their operating environment and continually adjust their parameters accordingly. This system has been proposed for achieving lower adaptation-delay and have area-delay-power efficient implementation. to Optimization has taken place from the synthesis, the area, delay and power of the proposed system. An attempt is made to design efficient architecture of adaptive filter.

## Index Terms- Adaptive filter, Area efficient, Partial product Generator

# I. INTRODUCTION

Nowadays one of the most widely used adaptive filter is LMS adaptive filter, because of its simplicity and converge performance.

Jyoti Dhiman et al described [1] the comparison between adaptive filtering algorithms that is least mean square (LMS), Normalized least mean square (NLMS). Implementation aspects of these algorithms are, SNR and computational complexity analysis. These algorithms use less input delay and less output delay.

The convergence speed of the least mean square (LMS) is dependent on the eigen value of the spread of input signal correlation matrix according to B.Widrow et al [2].

Y.Yi.R.Woods et al [3], Proposed fine grained pipelined design of an adaptive filter, it supports high sampling frequency but with pipeline depth. L.D.Van and W.S.Feng [4], proposed an efficient architecture for DLMS adaptive digital filter based on a new tree-systolic processing element.

The filter is based on the least-mean-square algorithm, but due to the problems in implementation of the systolic array,

a modified algorithm, a special case of the delayed LMS (DLMS), is used by H. Herzberg and R. Haimi-Cohen [5].

M. D. Meyer and D. P. Agarwal described [6] the direct-form LMS adaptive filter involves a long critical path due to an inner-product computation to obtain the filter output.

A transpose-form LMS adaptive filter is suggested by S. Ramanathan and V. Visvanathan in [7], where the filter output at any instant depends on the delayed versions of weights and the number of delays in weights varies from 1 to N.

Y.Yi et al [8] have proposed a fine-grained pipelined design to limit the critical path to the maximum of one addition time.

Van and Feng [9] have proposed a systolic architecture, where they have used relatively large processing elements (PEs) for achieving a lower adaptation delay with the critical path of one Multiply Accumulate operation.

Also higher power consumption proposed by L.-K. Ting et al [10], due to its large number of pipeline latches. Further effort has been made by Meher and Maheshwari [11] to reduce the number of adaptation delays.

P. K. Meher and S. Y. Park have proposed a 2-bit multiplication cell, and used that with an efficient adder tree. Also the implementation of pipelined inner-product concepts to minimize the critical path and silicon area without increasing the number of adaptation delays are explained in [12][13].

K. K. Parhi [14] explains how to design high-speed, lowarea, and low-power VLSI systems for a broad range of DSP applications.

The rest of this paper introduces the proposed system. Existing system is discussed in section II. Then, in section III, concentrates on the adaptive filter. Section IV deals with the simulation result of the paper. Finally Section V presents the conclusion of the paper.



p-ISSN: 2348-6848 e-ISSN: 2348-795X Volume 04 Issue 02 February 2017

#### II. RELATED WORK

The existing work on the DLMS adaptive filter does not discuss the fixed-point implementation issues, e.g., location of radix point, choice of word length, and quantization at various stages of computation, although they directly affect the convergence performance, particularly due to the recursive behavior of the LMS algorithm. Therefore, fixedpoint implementation issues are given adequate emphasis in this paper. Besides, they have presented the optimization of our previously reported design to reduce the number of pipeline delays along with the area, sampling period, and energy consumption. The proposed design is found to be more efficient in terms of the power-delay product (PDP) and energy-delay product (EDP) compared to the existing structures.

The block diagram of the DLMS adaptive filter is shown in

Fig. 4, where the adaptation delay of *m*cycles amounts to the delay introduced by the whole of adaptive filter structure consisting of finite impulse response (FIR) filtering and the weight-update process. It is shown in that the adaptation delay of conventional LMS can be decomposed into two parts: one part is the delay introduced by the pipeline stages in FIR filtering, and the other part is due to the delay involved in pipelining the weight update process.



Figure 1: Structure of the conventional delayed LMS adaptive filter.

#### III. PROPOSED SYSTEM

A new architecture design for the delayed LMS adaptive filter with 64 tap is proposed. Based on decomposition of delay on the existing system, the Delayed LMS adaptive filter can be implemented by a proposed structure shown in Fig. 2.



Figure 2: Structure of the modified delayed LMS adaptive filter.

Where, dn is the desired response, yn is the filter output, and en denotes the error computed during the nth iteration.  $\mu$  is the step-size, and N is the number of weights used in the LMS adaptive filter.

A)The proposed architecture is divided into two main components.

- 1. Error computation block
- 2. Weight update block Error computation block

The proposed architecture for error computation block for the delayed LMS adaptive filter is shown in fig 3. This architecture consists of delay element D, 2 bit PPG unit, adder tree with  $log_2N$  stages, and shift adder tree with  $log_2L$ -1 stages.



Figure 3: Proposed structure of the error-computation block.

# 1) 2-bit PPG unit:

This unit is generating the partial product values of the error computation block. This PPG consists of 2 to 3 decoder and AOC. The 2 to 3 decoder is generating the values of 1, 2 and



3 for the input of u1 and u0 when 01, 10, 11 respectively. The 2 bit PPG unit is shown in Fig 4.

The AOC is the AND/OR cell design. This is use to multiply the input data into the coefficient value. The architecture for AOC is shown in Fig 8. The structure and function of AND cells and OR cells are depicted by Fig. 5(b) and 5(c) respectively.



Figure 4: Proposed structure of PPG. AOC stands for AND/OR cell.



Figure 5: Structure and function of AND/OR cell. Binary operators • and + in (b) and (c) are implemented using AND and OR gates, respectively.

# 2) Architecture of adder tree:

Conventionally, It should have been performed the shift-add operation on the partial products of each PPG separately to obtain the product value and then added all the *N* product values to compute the desired inner product. However, the shift-add operation to obtain the product value increases the word length, and consequently increases the adder size of N – 1 additions of the product values. To avoid such increase in word size of the adders, we add all the *N* partial products of the same place value from all the *N* PPGs by one adder tree. The addition scheme for the error-computation block for a four-tap filter and input word size L=8 is shown in Fig. 6



Figure 6: Adder-structure of the filtering unit

# B)Weighted updated block:

The proposed structure for the weight-update block is shown in Fig. 7. It performs N multiply-accumulate operations of the form  $(\mu \times e) \times xi + wi$  to update N filter weights. The step size  $\mu$  is taken as a negative power of 2 to realize the multiplication with recently available error only by a shift operation. Each of the MAC units therefore performs the multiplication of the shifted value of error with the delayed input samples xi followed by the additions with the corresponding old weight values wi. All the Nmultiplications for the MAC operations are performed by NPPGs, followed by N shift- add trees. Each of the PPGs generates L/2 partial products corresponding to the product of the recently shifted error value  $\mu \times e$  with L/2, the number of 2-b digits of the input word xi where the sub-expression  $3\mu \times e$  is shared within the multiplier. Since the scaled error  $(\mu \times e)$  is multiplied with the entire N delayed input values in the weight-update block, this sub-expression can be shared across all the multipliers as well. This leads to substantial reduction of the adder complexity. The final outputs of MAC units constitute the desired updated weights to be used as inputs to the error-computation block as well as the weightupdate block for the next iteration.



Available at https://edupediapublications.org/journal Volu



Figure 7: Proposed structure of the weight-update block.

## IV. SIMULATION AND RESULT

The simulation of the proposed system architecture in Modelsim analysis the area, power, and delay of the proposed system in Spartan 6 using Xilinx software. The simulation result for adaptive filter is shows in Fig 8. The synthesis report of the adaptive filter is shown in Fig 9. Finally the comparison of the proposed system is detailed in Table 1.

The power consumption of proposed and existing systems as compared in above table 2. The existing system power consumption is around 0.24mW. but in case of proposed system power consumption is around 0.14mW. The proposed system is 64 tap fixed point filter. From this we can calculate energy per sample and also find energy delay product (EDP) by using different compliers.



Figure 8: Simulation result

Figure 9: Synthesis report

Table 1: Comparison

	Existing system	Proposed system
Filter length	32	64
Number of slices	4060	8631
Power	0.24	0.132
consumption(mW)		





Available at <a href="https://edupediapublications.org/journal">https://edupediapublications.org/journal</a>



Figure 11: Power consumption

## V. CONCLUSION

A Novel approach of high effective 64 tap fixed point DLMS adaptive filter is proposed. A partial product generator of multiplicants and inner product is used. The fixed point implementation scheme architecture with bit level clipping is proposed. From the synthesis, the area, delay and power of the proposed system is optimized. The proposed architecture gives the efficient output reuslts compared with existing ones. The existing system power consumption is 0.24mW but in case of proposed system power consumption is 0.132mW.It involved significantly less adaptation delay and provided significant saving of ADP and EDP compared to the existing structures Further we proceed pipelining implementation with partial product generator across the time consuming combinational blocks of filter structure.

# REFERENCES

- [1]. Jyoti dhiman, Shadab ahmad , Kuldeep Gulia, "Comparison between Adaptive filter Algorithms (LMS, NLMS and RLS)" International Journal of Science, Engineering and Technology Research (IJSETR) Volume 2, Issue 5, May 2013.
- [2]. B. Widrow, J. M. McCool, M. G. Larimore, and C. R. Johnson, Jr., "Stationary and nonstationary learning characteristics of the LMS adaptive filters," Proceedings of the IEEE, vol. 64, pp. 1151-1162, Aug. 1976.
- [3]. Y.Yi.R.Woods, L.-K. Ting, R.Woods and C.F.N. Cowan. 2005. "High speed FPGA based implementations of delayed – LMS filters," J. Very Large Scale Integr. (VLSI) Signal Process., vol. 39, No.1-2, pp. 113 – 131, Jan 2005.
- [4]. L. D. Van and W. S. Feng, "An efficient systolic architecture for the DLMS adaptive filter and its applications " IEEE Trans. Circuits Syst. II, Analog

Digital Signal Process., vol. 48, no. 4, pp. 359-366, Apr. 2001.

- [5]. H. Herzberg and R. Haimi-Cohen, "A systolic array realization of an LMS adaptive filter and the effects of delayed adaptation," *IEEE Trans. Signal Process.*, vol. 40, no. 11, pp. 2799–2803, Nov. 1992.
- [6]. M. D. Meyer and D. P. Agarwal, "A high sampling rate delayed LMS filter architecture," *IEEE Trans. Circuits Syst. II, Analog Digital Signal Process.*, vol. 40, no. 11, pp. 727–729, Nov. 1993.
- [7]. S. Ramanathan and V. Visvanathan, "A systolic architecture for LMS adaptive filtering with minimal adaptation delay," in *Proc. Int. Conf. Very Large Scale Integr. (VLSI) Design*, pp. 286–289.,Jan. 1996.
- [8]. Y. Yi, R. Woods, L.-K. Ting, and C. F. N. Cowan, "High speed FPGA-based implementations of delayed-LMS filter," *J. Very Large Scale Integr. (VLSI) Signal Process.* vol. 39, nos. 1–2, pp. 113–131, Jan. 2005.
- [9]. L. D. Van and W. S. Feng, "An efficient systolic architecture for the DLMS adaptive filter and its applications," *IEEE Trans. Circuits Syst. II, Analog Digital Signal Process.*, vol. 48, no. 4, pp. 359–366, Apr. 2001.
- [10]. L.-K. Ting, R. Woods, and C. F. N. Cowan, "Virtex FPGA implementation of a pipelined adaptive LMS predictor for electronic support measures receivers," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 13, no. 1, pp. 86–99, Jan. 2005.
- [11]. P. K. Meher and M. Maheshwari, "A high-speed FIR adaptive filter architecture using a modified delayed LMS algorithm," in *Proc. IEEE Int. Symp. Circuits Syst.*, pp. 121–124, May 2011.
- [12]. P. K. Meher and S. Y. Park, "Low adaptation-delay LMS adaptive filter part-I: Introducing a novel multiplication cell," in *Proc. IEEE Int. Midwest Symp. Circuits Syst.*, , pp. 1–4,Aug. 2011.
- [13]. P. K. Meher and S. Y. Park, "Low adaptation-delay LMS adaptive filter part-II: An optimized architecture," in *Proc. IEEE Int. Midwest Symp. Circuits Syst.*, pp. 1– 4,Aug. 2011,
- [14]. K. K. Parhi, VLSI Digital Signal Processing Systems: Design and Implementation. New York, USA: Wiley, 1999.