VLSI Implementation of Delayed LMS Adaptive Filter with Efficient Area-Power-Delay

Muthulakshmi.G, Revathi.S

Abstract*— In this paper, we present an efficient architecture for the implementation of a delayed least mean square Adaptive filter. For achieving lower adaptation-delay and area-delay-power, we use a novel partial product generator and propose an optimized balanced pipelining across the time-consuming combinational blocks of the structure. From synthesis results, we find that the proposed design with less area-delay product (ADP) and less energy-delay product (EDP) than the best of the existing systolic structures, for various filter lengths. We propose an efficient fixed-point implementation scheme in the proposed architecture. We present here the optimization of 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.

Index Terms*—Adaptive filters, Adder tree optimization, fixed-point arithmetic, least mean square (LMS) algorithms.

I. INTRODUCTION

The least mean square (LMS) adaptive filter is the most popular and widely used adaptive filter, because of its simplicity and its satisfactory convergence performance. The direct-form LMS adaptive filter involves a long critical path due to its inner-product computation to obtain the output from filter such that the critical path is required to be reduced by pipelined implementation when it exceeds to desired sample period. But the conventional LMS algorithm does not support for pipelined implementation because of its recursive behavior, so they are modified to a form called the delayed LMS (DLMS) algorithm, which allows pipelined implementation of the filter. A lot of work has been done to implement the DLMS algorithm in systolic architectures to increase the frequency but, they involve an adaptation delay

For filter length N, this is quite high for large order filters. We proposed a 2-bit multiplication cell, and with an efficient adder tree for pipelined inner-product computation to Minimize the critical path and silicon area without increasing the number of adaptation delays. The existing work on the DLMS adaptive filter does not discuss with the fixed-point implementation issues, such as the location of radix point, choice of word length, and quantization at various stages of computation. Therefore, fixed-point implementations in the proposed design 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.

Manuscript received January, 2014.

Muthulakshmi.G, M.E-Applied Electronics,, Dhanalakshmi Srinivasan College Of Engineering,Coimbatore,Tamilnadu,India.

Revathi.S, Assistant Professor, Dhanalakshmi Srinivasan College Of Engineering, Coimbatore, Tamilnadu, India.

II. ADAPTATION DELAY IN DLMS ADAPTIVE FILTR OVER CONVENTIONAL LMS ADAPTIVE FILTER

The block diagram of the DLMS adaptive filter is shown in Fig.1, shows 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. The adaptation delay of conventional LMS can be decomposed into two parts: first 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. Based on such a decomposition of delay, the DLMS adaptive filter can be implemented by a structure shown in Fig.2. The modified DLMS algorithm decouples the error-computation block and the weight-update block and allows performing optimal pipelining by feed forward cut-set retiming to minimize the number of pipeline stages and adaptation delay.

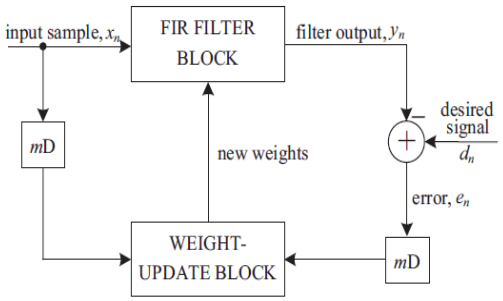

Fig.1 structure of conventional LMS adaptive filter

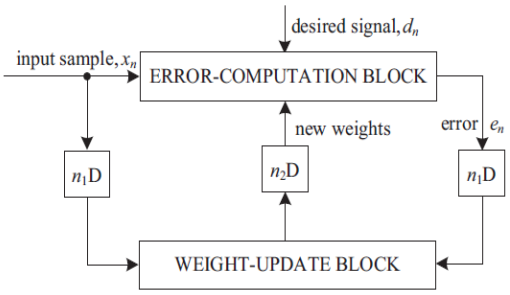

Fig.2 structure of delayed LMS adaptive filter

The adaptation delay gets reduced in DLMS due to its pipelined structure, but in conventional structure of DLMS, the systolic architectures are used such that there exist a high adaptation delay.

III. PIPELINED STRUCTURES OPTIMIZATION

A. Error-computation block

The proposed structure for error-computation unit of an N-tap DLMS adaptive filter is shown in Fig. 3. It consists of N number of 2-bit partial product generators (PPG) corresponding to N multipliers and a cluster of L/2 binary adder trees,

followed by a single shift–add tree. Each sub block is described in detail.

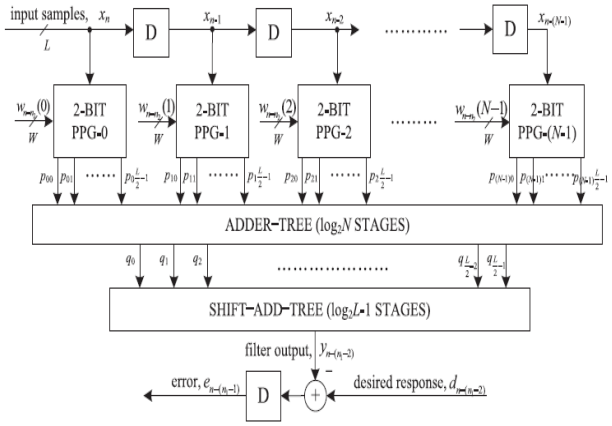

Fig.3 structure of error-computation block

1) Structure of PPG: The structure of each partial product generator (PPG) is shown in Fig.4.It consists of L/2 number of 2-to-3 decoders and the equal number of AND/OR cells (AOC) Each of the 2-to-3 decoders takes a 2-bit digit (u1u0) as input and produces three outputs b0 = u0. u1, b1 = u0 · u1, and b2 = u0 · u1, such that b0 = 1 for (u1u0) = 1, b1 = 1 for (u1u0) = 2, and b2 = 1 for (u1u0) = 3. The decoder output b0, b1 and b2 along with w, 2w, and 3w are given to an AOC, where w, 2w, and 3w are in 2's complement representation and sign-extended to have (W + 2) bits each.

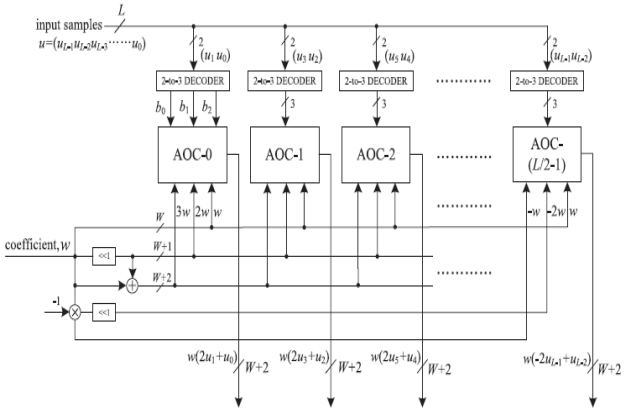

Fig.4 structure of partial product generator

2) Structure of AOCs: The structures of an AOC consists of three AND cells and two OR cells. Each AND cell takes an n-bit input and a single bit input b, also consists of n AND gates. It distributes all the n bits of input D to its n AND gates as one of the inputs. The other inputs of all the n AND gates are fed with the single-bit input b. The output of an AOC is w, 2w, and 3w corresponding to the decimal values 1, 2, and 3 of the 2-b input (u1u0). The decoder along with the AOC performs 2-bit multiplication and L/2 parallel multiplications with a 2-bit digit to produce L/2 partial products of the product word.

3) Structure of Adder Tree: The shifts-add operation on the partial products of each PPG gives the product value and then added all the N product values to compute the inner product output. However, the shift-add operation obtains the product value which increases the word length, and the adder size. To avoid 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 a single adder tree. Table I shows the pipeline latches for various filter lengths.

Table I: Location of pipeline latches for L=8, N=8,16and32

N	Error-computation block		Weight-Update Block
	Adder Tree	Shift-add Tree	Shift-add Tree
8	Stage-2	Stage-1 and 2	Stage-1
16	Stage-3	Stage-1 and 2	Stage-1
32	Stage-3	Stage-1 and 2	Stage-2

B. weight-update block

The proposed structure of weight-update block is shown in Fig. 5. It performs N multiply-accumulate operations of the form $(\mu \times e) \times xi + w_i$ to update N filter weights. The step size μ is taken as a negative power of 2 to realize the multiplication with recently available error by the shift operation. Each MAC unit 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 w_i. All the MAC operations are performed by N PPGs, 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 μ × e with the number of 2-bit digits of the input word x_i. The sub expression can be shared across all the multipliers. This leads to a gradual reduction adder in complexity. The final outputs of MAC units constitute updated weights to be used as inputs to the error-computation block and the weight-update block for the next iteration.

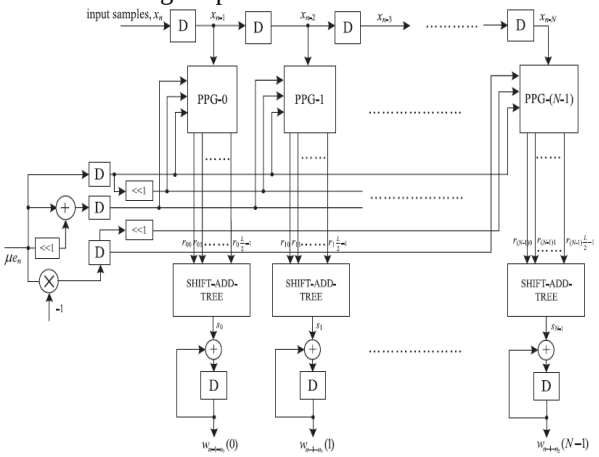

Fig.5 structure of weight-update block

C. Fixed-point considerations

The fixed-point implementation of the proposed DLMS adaptive filter shows the bit level pruning of the adder tree, to reduce the hardware complexity without the degradation of steady state MSE. For fixed-point implementation, the word lengths and radix points for input samples, weights, and internal signals are need to be decided. Fig. 6 shows the fixed-point representation of a binary number. Table II shows the fixed-representation of the desired signals; its quantization is usually given as an input. For this purpose, the specific scaling/sign extension and truncation/zero padding are required.

Since the LMS algorithm performs learning so that y has the same sign as d, the error signal e can also be set to have the same representation as y without overflow after the subtraction.

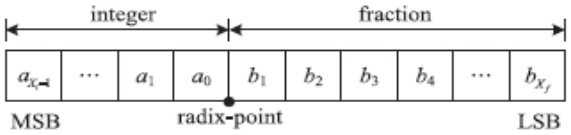

Fig.6 fixed-point representation of a binary number

Table II.Fixed-point representation of the signals of the proposed DLMS adaptive filter.

Signal Name	Fixed-Point Representation
X	(L, L_i)
W	(w, w_i)
P	(w+2, w_i+2)
Q	(w + 2 + $\log_2$N , w_i +2 + $\log_2$N)
y, d, e	(w, w_i+ L_i+ $\log_2$N)
µe	(w, w_i)
R	(w+2, w_i+2)
S	(w, w_i)

IV. ADDER TREE OPTIMIZATION

The adder tree and shift–add tree computation can be pruned for further optimization of area, delay, and power complexity. The adder tree structure is given in fig.7. To reduce the computational complexity, some of the LSBs of inputs of the adder tree can be truncated and the guard bits can be used to minimize the impact of truncation on the error performance of the adaptive filter. To have more hardware saving, the bits to be truncated are not generated by the PPGs, so the complexity of PPGs also gets reduced. To have more hardware saving, the bits to be truncated are not generated by the PPGs, so the complexity of PPGs also gets reduced.

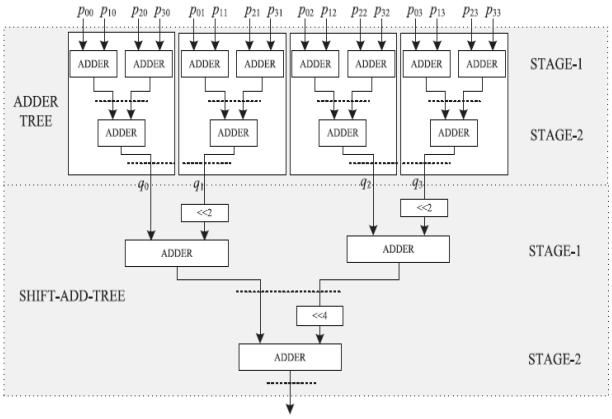

Fig.7 structure of adder tree

V. RESULT ANALYSIS

The Simulation results are carried out for Fixed-point LMS adaptive filter to find out the low adaptation delay. The Simulation is carried out by the Modelsim 6.3f as a simulator tool. The performance of the delay block is simulated by giving various inputs to the weight-update block with various weight is given below. The error can be estimated from the various iterations which are shown below. The Simulation Model & its waveform for delay with its weights w1, w2, w3, w4 is given below in fig 8.

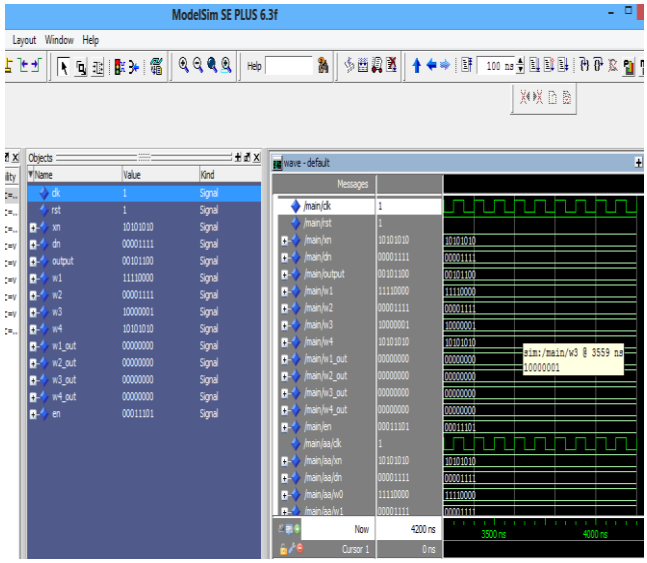

Fig 8: simulation model for delay with Xn=10101010, Dn=00001111, Rst=1 And Clk=1

The Simulation Model & its waveform for delay with its weights w1, w2, w3, w4 is given below in fig 9.

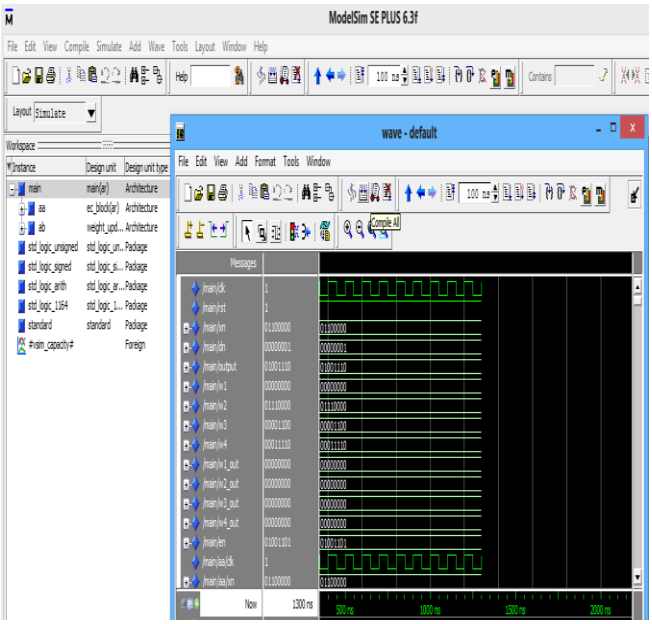

Fig 9: simulation model for delay with Xn=01100000, Dn=00000001, Rst=1 And Clk=1

The Simulation Model & its waveform for delay with its weights w1, w2, w3, w4 is given below in fig.10.

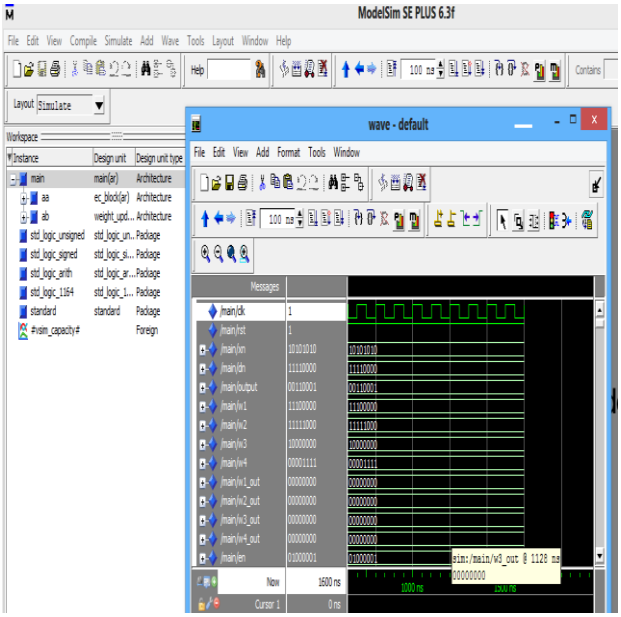

Fig 10: simulation model for delay with Xn=10101010, Dn=11110000, Rst=1 And Clk=1

VI. CONCLUSION

An area–delay-power efficient with low adaptation-delay architecture for fixed-point implementation of DLMS adaptive filter are achieved by using a novel PPG for efficient implementation of general multiplications and inner-product computation by common sub expression. From this, proposed strategy an optimized balanced pipelining across the time-consuming blocks is to reduce the adaptation delay and power consumption. The proposed structure involved significantly less adaptation delay and provided significant saving of ADP and EDP compared to the existing structures.

FUTURE SCOPE

The efficient addition scheme reduces the adaptation delay to achieve the faster performance and reduction in the critical path supports the high input-sampling rates. The future work involves that to reduce the adder complexity by replacing the various adders in adder blocks to achieve the better performance of area and power. The fixed-point implementation of the proposed architecture derives the expression for steady-state error.

REFERENCES

1. B. Widrow and S. D. Stearns, Adaptive Signal Processing Englewood Cliffs, NJ, USA: Prentice-Hall, 1985.
2. S. Haykin and B. Widrow, Least-Mean-Square Adaptive Filters Hoboken, NJ, USA: Wiley, 2003.
3. M. D. Meyer and D. P. Agrawal, "A modular pipelinedImplementation of a delayed LMS transversal adaptive Filter," in Proc. IEEE Int. Symp. Circuits Syst., May 1990,pp. 1943–1946.
4. G. Long, F. Ling, and J. G. Proakis, "The LMS algorithmWith delayed coefficient adaptation," IEEE Trans.Acoust. Speech, Signal Process. vol.37, no. 9, pp.1397–1405, Sep. 1989.
5. G. Long, F. Ling, and J. G. Proakis, "Corrections to 'The LMS algorithm with delayed coefficient adaptation'," IEEE Trans. Signal Process.,vol. 40, no. 1, pp. 230–232,Jan. 1992.
6. H. Herzberg and R. Haimi-Cohen, "A systolic arrayRealization of an LMS adaptive filter and the effects of delayed adaptation," IEEE Trans.Signal Process., vol. 40,no. 11, pp. 2799–2803, Nov. 1992.
7. M. D. Meyer and D. P. Agrawal, " IEEE Trans. Circuits Syst. II, Analog Digital Signal Process, vol. 40, no. 11, pp. 727–729, Nov. 1993.
8. S. Ramanathan and V. Visvanathan, "A systolic architecture for LMS adaptive filtering with minimal adaptation delay," in Proc. Int. Conf. Very Large ScaleIntegr. (VLSI) Design, Jan. 1996, pp. 286–289.
9. Y. Yi, R. Woods, L.-K. Ting, and C. F. N. Cowan, "High Speed FPGA- based implementations of delayed- LMS filters," J . Very Large Scale Integr. (VLSI) Signal Process., vol. 39, nos. 1–2, pp. 113–131, Jan. 2005.
10. L. D. Van and W. S. Feng, "An efficient systolic architecture for the DLMS adaptive filter and its applications," IEEE Trans. Circuits Syst. II, AnalogDigital Signal Process., vol. 48, no. 4, pp. 359–366, Apr. 2001.
11. L.K. Ting, R. Woods, and C. F. N. Cowan, "VirtexFPGA implementation of a pipelined adaptive LMSPredictor for electronic support measures receivers," IEEE Trans. Very Large Scale Integr. (VLSI) Syst.,vol. 13, no. 1, pp. 86–99, Jan. 2005.
12. 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.,May 2011, pp. 121–124.
13. P. K. Meher and S. Y. Park, "Low adaptation-delay LMSAdaptive filter part-I: Introducing a novel multiplicationcell," in Proc. IEEE Int. Midwest Symp. Circuits Syst.,Aug. 2011, pp. 1–4.
14. 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., Aug.2011, pp. 1–4.
15. K. K. Parhi, VLSI Digital Signal Processing Systems: Design and Implementation. New York, USA: Wiley, 1999.
16. C. Caraiscos and B. Liu, "A roundoff error analysis of theLMS adaptive algorithm," IEEE Trans. Acoust., Speech, Signal Process., vol. 32, no. 1, pp. 34–41, Feb. 1984.
17. R. Rocher, D. Menard, O. Sentieys, and P. Scalart, "Accuracy evaluation of fixed-point LMS algorithm," inProc. IEEE Int. Conf. Acoust., Speech, Signal Process.,May 2004, pp. 237–240.

AUTHORS PROFILE

Muthulakshmi.G, received the B.E. degree in Electronics & communication engineering from Bharath Niketan engineering college under the Anna University, Chennai, 2008 to 2012. I am currently pursing M.E-Applied Electronics in Dhanalakshmi Srinivasan College of Engineering, Coimbatore. My area of interest are Advanced Digital Signal Processing, low power VLSI, Embedded systems.

Revathi.S, received the B.E. degree in Electronics & communication engineering from Muthayammal engineering college under the Anna University, Chennai, 2006 to 2010. I am completed M.E-communication systems in S.A Engineering College, Chennai. My area of interest are low power VLSI, signal integrity design for high speed digital system.