ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Issue 1, January 2016

moo

FIXED POINT IMPLEMEMNATION OF ADAPTIVE FILTER

R.Mythili

Department of ECE

Karpagam college of engineering

Coimbatore, India.

rmythilichandran@gmail.com

Dr.B.Nagaraj

HOD department of ECE

Karpagam college of engineering

Coimbatore, India.

nagarajice@gmail.com

ABSTRACT: The least mean square (LMS) adaptive filter is the most popular and most widely used adaptive filter, because of its simplicity and superior convergence performance .Since the conventional LMS algorithm does not support pipelined implementation because of its repetitive behaviour, it is transformed to a form called delayed LMS (DLMS) algorithm, which supports the pipelined implementation of the filter .This paper presents an efficient architecture for the implementation of a delayed least mean square Adaptive filter. For achieving lower adaptation-delay and area-delay power, for that we use a novel partial product generator and propose an optimized balanced pipelining across the time-consuming combinational blocks of the structure. It is 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. Hence it is clear that the total area-power consumption can be decreased to a great extent using the proposed method.

INDEX TERMS: LMS Adaptive filters, fixed point arithmetic, least mean square (LMS) algorithms, area efficient, low power.

I .INTRODUCTION:

The direct-form LMS adaptive filter involves a long critical path due to an inner-product computation to obtain the filter output. The critical path is needed to be reduced by pipelined implementation when it exceeds the desired sample period. The LMS algorithm does not support pipelined implementation, a variation of the Least Mean Squares (LMS) called delayed LMS (DLMS) algorithm which ideally suited for highly pipelined, adaptive digital filter implementation. A lot of work has been done to implement the DLMS algorithm in most efficient way. In this paper we proposed a 2-bit multiplication cell and with an efficient adder tree for pipelined inner product computation to minimize the critical path and area without increasing the number of adaption delays. The existing work on DLMS does not discuss on the fixed point implementation issues, such as location of radix point, choice of word length, quantisation 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 proposed to reduce the adaptation delay**.**

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

Fig.1 shows the block diagram of the DLMS adaptive filter, 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

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Issue 1, January 2016

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

.

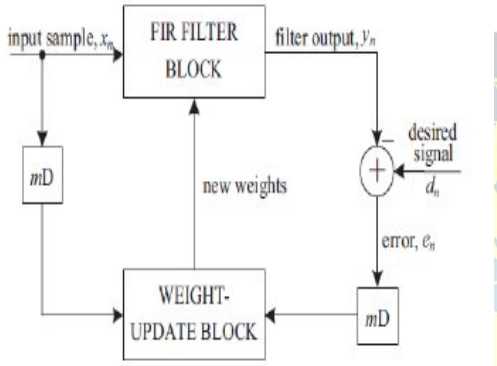

Fig.1 Structure of conventional LMS adaptive filter

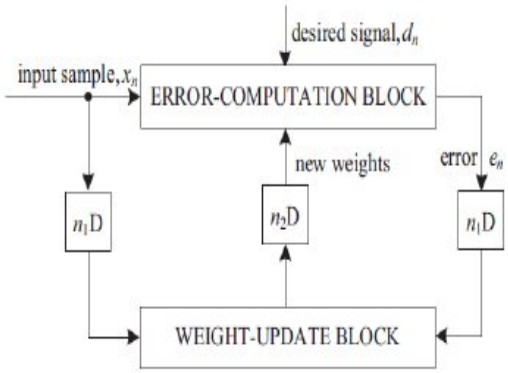

Fig.2 Structure of delayed LMS adaptive filter

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

III. OPTIMIZATION IN PIPELINED STRUCTURES

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 L/2 binary adder trees, then followed by a single shift–add tree. Each sub block is described in detail

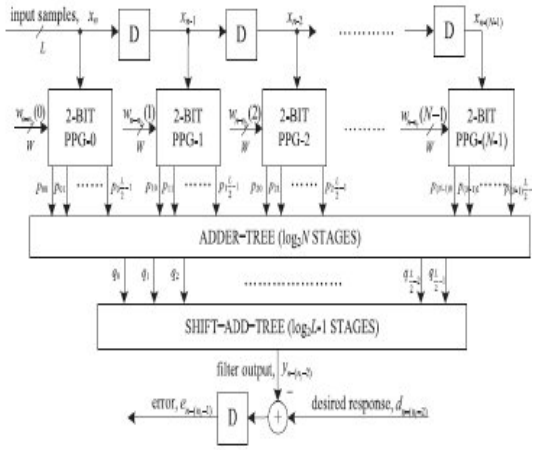

Fig.3 Structure of error-computation block

1)Structure of PPG: The structure of partial product generator (PPG) is shown in Fig.4.It consists of L/2 number of 2-to-3 decoders and of AND/OR cells (AOC). Each of the 2-to-3 decoders takes a 2-bit digit (u1u0) as its input and produces b0 = u0. u1, b1 = u0 · u1, and b2 = u0· u1 these three outputs, 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 as input to an AOC, where w, 2w, and 3w are in 2's complement representation

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Issue 1, January 2016

and sign-extended to have (W + 2) bits each

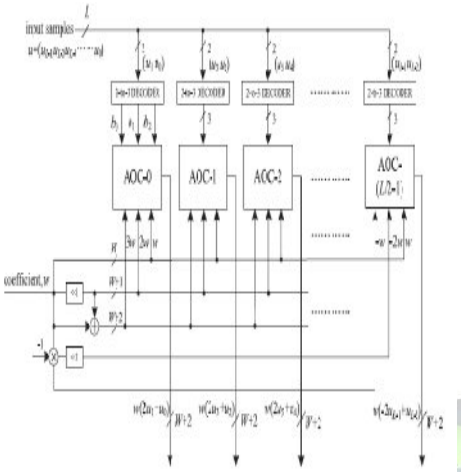

Fig.4 structure of partial product generator

2) Structure of AOCs*:* three AND cells and two OR cells are present in an AOCs. Each AND cell takes an n-bit input and a single bit input b, there present an n number of AND gates. Each AND gate act as one of the input and it distributes all the n bit of D input.The remaining 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 performs 2-bit multiplication and L/2 parallel multiplications with a 2-bit digit to produce L/2 partial products of the product word along with an AOC block.

3) Structure of Adder Tree: The structure of adder tree is shown which performs shifts-add operation on the partial products of each PPG gives the product value and then add all the N product values to compute the inner output of the product. However, this shift-add operation obtains the product value which increases the word length, and also the adder size. To avoid the problem of increasing word size of the adders, we going to 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, 16 and 32

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 Fig. 5 shows the proposed structure of weight-update block. It performs N multiply-accumulate operations of the form (μ × e) × xi + wi 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 is used to 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. The N PPGs perform all the MAC operation,and then 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 xi. The sub expression can be shared across all the multipliers. This results to a gradual reduction in adder complexity. The final outputs of MAC units is used to updated weights to be used as inputs to the error-computation block and the weight-update block for the next iteration.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Issue 1, January 2016

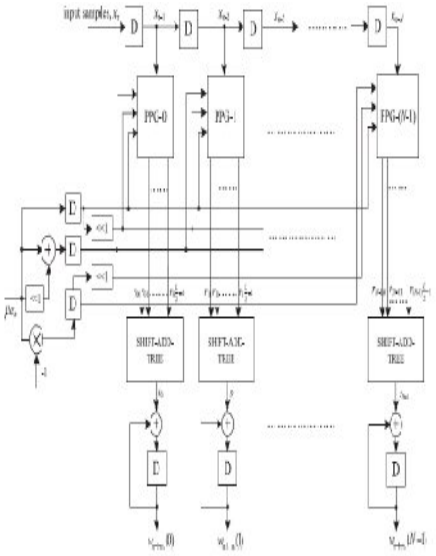

Fig.5 structure of weight-update block

C. Fixed-point considerations

The fixed-point implementation of the proposed DLMS adaptive filter shows the bit level cut back 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.

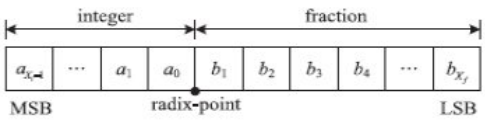

Fig.6 fixed-point representation of binary number.

Table II. Fixed-point representation then of the signals.

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

The adder tree and shift –add tree computation can be cut back for further optimization of area, delay and power complexity.fig 7 shows the adder tree .in order to reduce the complexity in computation some of the least bits are truncated and the guard bits are used to reduce the truncation error which occur in adaptive filter. since to reduce the hardware usage the truncated bits are not generated by the PPGs. This further reduces the PPGs complexity

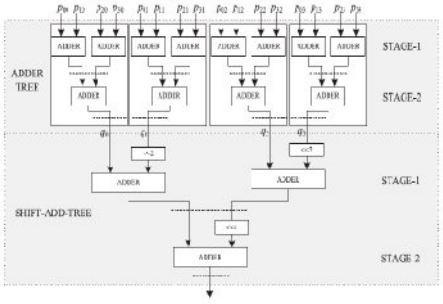

Fig.7 structure of adder tree

This can be done using carry save adder ,which is nothing but a type of digital adder, used in computer micro architecture to compute the sum of three or more *n*-bit numbers in binary. It differs from other digital adders in that it outputs two numbers of the same dimensions as the inputs, one which is a sequence of partial sum bits and another which is a sequence of carry bits.With the help of carry save adder complexity of addition can be reduced.

V. RESULT ANALYSIS

The Simulation results are carried out for Fixed-point LMS adaptive filter using carry save adder to

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Issue 1, January 2016

find out the low adaptation delay. The Simulation is carried out by the Xilinx 13.2 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.

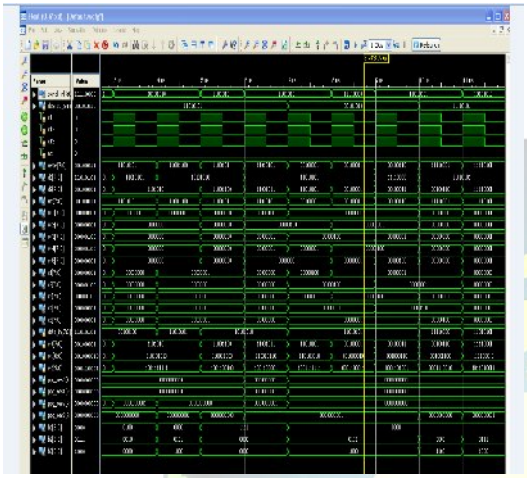

Fig 8: simulation result of filter block

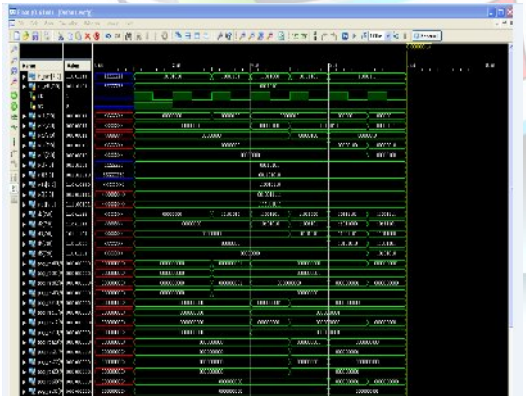

Fig 9: simulation result of weight-update block

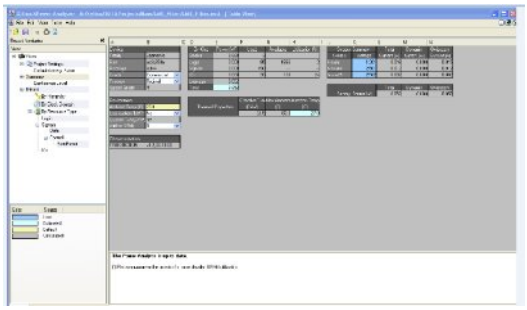

Fig 10: simulation result of low power consumption

VI. CONCLUSION

This paper is based on efficient implementation of adaptive filter to achive faster performance and reduction in the critical path supports the high input-sampling rates .The adaption delay get reduced by using fixed point implementation of DLMS adaptive filter by using a novel PPG for implementation of general multiplications and inner-product computation by carry save adder . . 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 significantly reduces adaptation delay and provided significant saving of ADP and EDP compared to the existing structures.

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.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Issue 1, January 2016

[7] M. D. Meyer and D. P. Agrawal, "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.
[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.