Design and Implementation of Low Complexity LMS Adaptive Filter

Dr P Kalpana[3]
{kp.ece@psgtech.ac.in[1]}

Student, ME VLSI, ECE, PSG College of Technology, Coimbatore[1], Assistant Professor(Sel Gr), Department of ECE, PSG College of Technology, Coimbatore[2], Professor, Department of ECE, PSG College of Technology, Coimbatore[3]

Abstract. Adaptive filter is a computational device that iteratively models the relationship between the input and output signals of a filter in the real time. It works based on an adaptive algorithm that self-adjusts the coefficients of the linear filter iteratively to minimize the power of e(n). The LMS algorithm is one of the most popularly used adaptive algorithm among others which adjusts the coefficients of adaptive filters. There are two main computing blocks in the direct-form LMS adaptive filter, namely, the error-computation block and the weight-update block which decides the efficiency of the filter. In this paper, adaptive filter is implemented in two different architectures namely, zero adaptation delay adaptive filter and two adaptation delay adaptivefilter which result in low power consumption and less area complexity.Zero adaptation delay adaptive filter provides nearly 52% savings in area and the delay decreases by 26% in two adaptation delay adaptive filter over the conventional adaptive filter. Hence based on the required speed and area for the application, any one of the proposed structures can be used.

Keywords: Adaptive Filter, Least Mean Square Algorithms, LMS Adaptive Filter, Adaptation Delay, Area, Delay.

1 Introduction

Adaptive filters are used in areas like signal prediction, signal enhancement, channel equalization, system identification and noise cancellation etc. However real time operations are required in filtering applications. Adaptive filter self-adjust its transfer function according to an optimizing algorithm and accuracy can be achieved by modification of its characteristics[1].An adaptive filter can be categorized into two components: a filter to process the input signal x(n) and its adaptive algorithm to update the coefficient weights of the filter. The block diagram of the adaptive filter is shown in Fig. 1 where d(n) represents the desired response, y(n) represents the output response of the filter ande(n) represents the error difference between d(n) and y(n). The adaptive algorithm will update the coefficient weights of the filter until output response of the filter is equal or nearly equal to the desired response[2].

ICCAP 2021, December 07-08, Chennai, India

DOI 10.4108/eai.7-12-2021.2314960

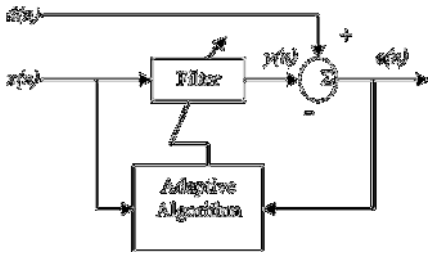

Fig 1 Block Diagram of adaptive filter

Adaptive algorithms are used to adjust the weights of adaptive equalizers towards an optimum configuration to minimize the error in the signal. It is selected based on the performance measures like rate of convergence, misadjustment, level of complexity and numerical properties etc. Zero forcing (ZF), least mean squares (LMS), Recursive Least Square (RLS), Kalman, and Least Square Lattice (LSL) are examples of algorithms used for adaptive equalization[3]. Due to simple operation, fast convergence,low complexity and superior performance, LMS is the most widely used algorithm for adaptive filters. A fast convergence indicates that the adaptive filter takes less time to calculate the filter coefficients.

The LMS algorithm is implemented by updating the filter weights using the estimated error during each sampling period[4]. Estimated error is given by difference between the desired response and the current filter output. The LMS algorithm performs the following operations to update the coefficients of an adaptive FIR filter:

1. Calculates the output signal y(n) from the FIRfilter

$$y(n) = \sum_{j=1}^{p} \widehat{w_j}(\mathrm{n}) x_j(n) \qquad (1)$$

2. Updates the filter coefficients by using the followingequation:

$$W(n+1) = \widehat{w}(n) + \eta[d(n) - x^T(n)\widehat{w}(n)]x(n) \qquad (2)$$

Here η represents the step size, $\widehat{w}(n)$ is the filter coefficient, x(n) is the input signal to the filter and n is the number of samples. Since the constraints on area, time and performance complexity is increasing there is a need for efficient implementation of LMS adaptive filter. This paper introduces the implementation of the LMS finite impulse response (FIR) adaptive filter by using Verilog and the performance indicators of adaptive filters.

The proposed architecture has an efficient addition scheme for error computation block and weight update block, to reduce the adaptation delay efficiently in order to achieve faster convergence performance and to reduce the critical path to support high sampling rates.

In the next section, the structure of conventional adaptive filter is given and we have described the proposed architectures in section III. Section IV shows the simulation results of conventional filter and proposed architectures with the comparison in area and delay followed by conclusion in section V.

2 Conventional Adaptive Filter

The conventional adaptive filter structure for a direct-form LMS adaptive filter consists of N multipliers. The multiplier is having two inputs, one input is from the common tap delay line which consists of delayed input signals x(n-D) and the other input is from 2:1multiplexer. The multiplexer will get the inputs from error computation block i.e. the estimated error value multiplied by the step size and from weight update block that is updated coefficient values[5]. The proposed structure also consists of N adders for calculating N weights. The adder tree gives the output of the filter by adding the outputs from N multipliers. This structure also requires a subtractor to calculate the error difference between obtained response and desired response and 2:1 de-multiplexers to transfer the product values to the adder tree block or to the weight update block. The clock signal is used to control the multiplexers and de-multiplexers [6].

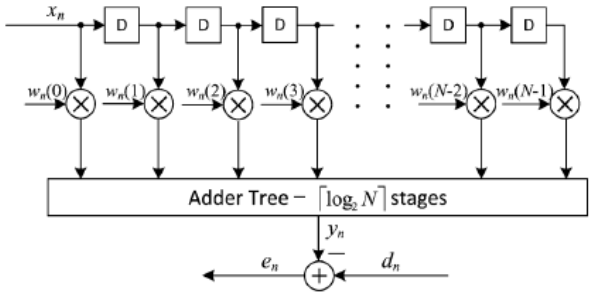

Fig. 2(i). Error Computation Block

The conventional adaptive filter consists of two main blocks called error computation block and weight update blocks which are shown in Fig. 2(i) and (ii). After multiplication of error value with the step size in error computation block, a pipelined latch is introduced .The multiplication is done by hardwired shift method since step size is assumed to be a fraction of 2. So, there is no register overhead in pipelining. Both the blocks will share the same set of registers.

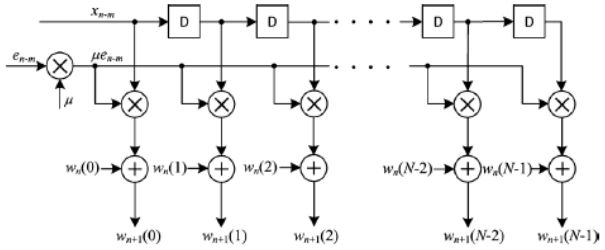

Fig. 2(ii). Weight Update Block

3 Implementation Of Proposed Structures

Zero Adaptation Delay Adaptive Filter

Zero adaptation delay denotes that the computation of error value and weight updating takes place at the same clock cycle. This cannot occur concurrently because the structure is non-pipelined. Zero adaptation delay can be achieved by making the blocks to complete its operation within half clock cycle and multiplexing the same set of multipliers. Same set of registers are used for storing the computed values in both the blocks. The Structure of zero adaptation delay adaptive filter is shown in Fig 3.

The structure is designed in such a way that it should take the new input sample in every clock cycle. So, at each rising edge of the clock, the registers present in the delay path will process the input signal and remains unchanged for the entire clock period. The multiplexers transfer the weight values stored in registers to multiplier during the first half of each clock period. After multiplication, the multiplier passes the product words to adder tree to calculate the filter output through the de-multiplexers and the subtractor computes the error value. This computed error value is broadcasted to all multipliers in weight update block after a right shift operation.

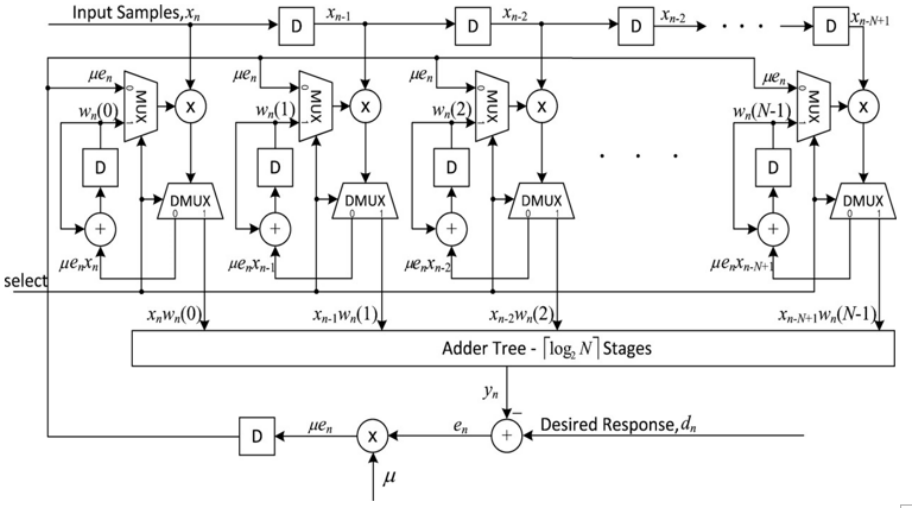

Fig. 3. Structure of zero adaptation delay adaptive filter

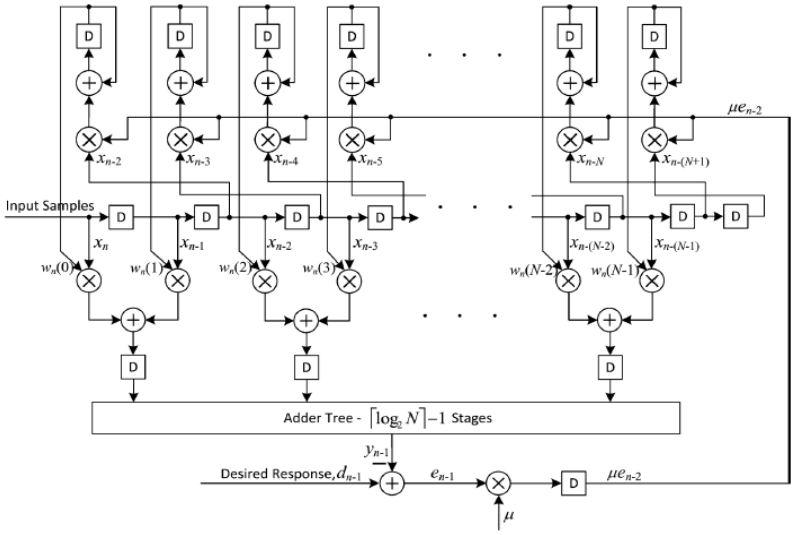

Fig. 4. Structure of two adaptation delay adaptive filter

The select line of the multiplexer and de-multiplexer decides the functionality of the filter. If it is zero, the filter will update the coefficient values else it will work as normal FIR filter. Since the error value is multiplied by the step size in error computation block and given to weight update blocks, the adaptation time of the filter is totally decided by the performance of those two blocks. The weight updation takes lesser time when compared to error computation, but we can't predict when the blocks will complete its working. Therefore, the structure is implemented in such a way that error computation is done first and weight updation is done at the second half cycle [7]-[9].

Two Adaptation Delay Adaptive Filter

The structure of proposed two adaptation delay LMS adaptive FIR filter is shown in Fig. 4. It is implemented with three pipeline stages. The first level of the adder tree in the error computation block comes in the first pipeline stage. The next pipeline stage comprises the rest of the block. The third pipeline stage is made in the weight update block. Compared to conventional adaptive filter, this structure requires more registers because of pipelining. This structure has N multipliers and N adders for calculating weights[10]-[11]. An adder tree is constructed for adding the output of multipliers to produce the filter output.The proposed structure also requires a subtractor to calculate the error value and 2:1 de-multiplexers to transfer the product values to the adder tree block or to the weight update block as in zero adaptation delay adaptive filter[12]. The clock signal is used to control the multiplexers and de-multiplexers.

4 Simulation Results

The conventional LMS Adaptive filter and the proposed structures are implemented using Verilog HDL. This design is simulated using Modelsim 6.4c and Synthesized by Xilinx

13.2/9.1[13]-[14].The comparison is made on area and delay of conventional, zero adaptation delay and two adaptation delay adaptive filters.The simulation waveform of conventional adaptive filter is shown in Fig. 6. From the waveform, it is clear that if reset goes high then the error value and expected response value is same since the output value is zero. If reset goes low, the filter will calculate the error value and update the filter coefficient value with one adaptation delay that is with one clock cycle delay.

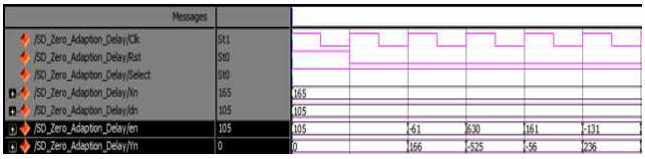

Fig. 5. Simulation result of Conventional adaptive filter

Thesimulation result of zero adaptation delay adaptive filter is shown in Fig. 6. From the waveform, it is inferred that if reset is zero, the filter will calculate the error value and update the filter coefficient value without any delay. Hence the structure is called zero adaptation delay adaptive filter.

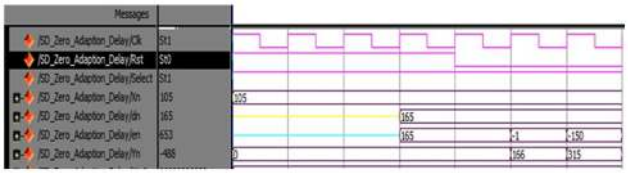

Fig. 6.Simulation result of zero adaptation delay adaptive filter

The simulation result of conventional adaptive filter is shown in figure 8. This structure calculates the error value and updates the filter coefficients with delay of two clock cycles. Hence the structure is called two adaptation delay adaptive filter.

Fig. 7.Simulation result of two adaptation delay adaptive filter

Table I shows the comparison between conventional, zero adaptation delay and two adaptation delay adaptive filters. From the comparison table, it is inferred that hardware requirement for zero adaptation delay adaptive filter is lesser than two adaptation delay adaptive filter at the cost of increase in delay.

Table I Comparison on area and delay

Method Name	Area			Delay		
	Gates	Slice FF	LUT	Max Delay (ns)	Gate Delay (ns)	Path Delay (ns)
Proposed Structures						
A. Zero Adaptation Delay[6]	17,253	78	165	21.205	13.879	7.125
B. Two Adaptation Delay[6]	33,017	98	99	17.241	13.118	4.288
Conventional Adaptation Delay[6]	33,441	33	110	23.303	16.855	6.448

5 Conclusion

Based on the area and delay analysis the low-complexity architectures for the LMS adaptive filter are proposed. The direct form LMS adaptive filter needs very less registers than the transpose-form adaptive filter so it converges faster than the transpose form adaptive filter, but the drawback is delayed weight adaptation and large area requirement. To overcome this, two different architectures of direct form LMS adaptive filter with i) zero adaptation delay adaptive filter, ii) two adaptation delay adaptive filter are proposed. Zero adaptation delay adaptive filter does not have any adaptation delay. It has the minimum area requirement when compared with the two adaptation delay adaptive filter, but the drawbackis extra select pin and this architecture will not update the coefficient value automatically. So, it can be used in the areas where noise is very less like noise cancelling in AC electrical measurements.In two adaptation delay adaptive filter, path delay is very less and it will automatically update the coefficient value based on the error, so no need of extra pins to update the coefficient values. The weight update block in this structure will update the weight in two clock cycle. It can be used in application which require accurate data like noise removal in ECG signals.

References

[1] W. A. Harrison, J. S. Lim, and E. Singer, “A new application of adaptive noise cancellation,” IEEE Trans. Acoust., Speech, Signal Process., vol. 34, no. 1, pp. 21–27, Feb. 1986.

[2] R. H. Hearn, J. R. Zeidler, E. Dong, Jr., and R. C. Goodlin, “Adaptive noise cancelling: Principles and applications,” Proc. IEEE, vol. 63, no. 12, pp. 1692–1716, Dec. 1975.

[3] Harjeet Kaur and Dr.Rajneesh Talwar, “Performance Comparison of Adaptive Filter Algorithms for Noise Cancellation” IEEE Signal Process Letters., vol.16, no. 1, Jan. 2009.

[4] 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.G. Long, F. Ling, and J. G. Proakis, “The LMS algorithm with delayed coefficient adaptation,” IEEE Trans. Acoust., Speech, Signal Process., vol. 37, no. 9, pp. 1397– 1405, Sep.1989.

[5] M. D.Meyer and D. P. Agrawal, “Amodular pipelined implementation of a delayed LMS transversal adaptive filter,” in Proc. IEEE Int. Symp. Circuits Syst., May 1990, pp. 1943–1946.

[6] Pramod Kumar Meher, Sang Yoon Park, “Critical-Path Analysis and Low-Complexity Implementation of the LMS Adaptive Algorithm,” in IEEE Transaction on Circuits and Systems—I: regular papers,vol. 61, no. 3, March 2014

[7] S. Haykin and B. Widrow, Least-Mean-Square Adaptive Filters. Hoboken, NJ, USA: Wiley-Interscience, 2003.
[8] G. Long, F. Ling, and J. G. Proakis, "The LMS algorithmwith delayed coefficient adaptation," IEEE Transaction on Acoustic, Speech, Signal Processing, vol. 37, no. 9, pp. 1397–1405, Sep. 1989.
[9] L. D. Van and W. S. Feng, "An efficient systolic architecture for the DLMS adaptive filter and its applications," IEEE Transaction on Circuits System II, Analog Digital Signal Processing, vol. 48, no. 4, pp. 359–366, Apr. 2001.
[10] L.-K.Ting,R.Woods andC. F. N. Cowan, "Virtex FPGAimplementation of a pipelined adaptive LMS predictor for electronic support measuresreceivers," IEEE Transaction on Very Large Scale Integration, vol. 13, no. 1, pp. 86–99, Jan. 2005.
[11] E. Mahfuz, C. Wang, and M. O. Ahmad, "A high-throughput DLMS adaptive algorithm", IEEE International Symposium on Circuits System, pp. 3753–3756,May 2005.
[12] G. A. Clark, S. K. Mitra, and S. R. Parker, "Block implementation of adaptive digital filters," IEEE Trans. Acoust., Speech, Signal Process.,vol. ASSP-29, no. 3, pp. 744–752, Jun 1981.
[13] S. Y. Park and P. K. Meher, "Low-power, high-throughput, and lowarea adaptive FIR filter based on distributed arithmetic," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 60, no. 6, pp. 346–350, Jun. 2013.
[14] https://www.xilinx.com