ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

An Efficient Fixed Point LMS Adaptive Filter with Less Area and Low Delay

A.Geethavani
PG Scholar,
Department of ECE,
Intellectual Institute of Technology, AP, India.

P. Kullaya Swamy
Assistant Professor,
Department of ECE,
Intellectual Institute of Technology, AP, India.

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.

1.INTRODUCTION:

The filter is an important component in the communication world. It can eliminate unwanted signals from useful information.However, to obtain an optimal filtering performance, it requires 'a priori' knowledge of both the signal and its embedded noise statistical information. The classical approach to this problem is to design frequency selective filters, which approximate the frequency band of the signal of interest and reject those signals outside this frequency band.The removal of unwanted signals through the use of optimization theory is becoming popular, particularly in the area of adaptive filtering.

These filters minimize the mean square of the error signal, which is the difference between the reference signal and the estimated filter output, by removing unwanted signals according to statistical parameters. The Least Mean Square (LMS) adaptive filter is the widely used filter because of its simplicity and performance. The Least Mean Square adaptive filter is used here because it differs from a traditional digital filter in many ways-A traditional digital filter has only one input signal x(n) and one output signal y(n). An adaptive filter requires an additional input signal d(n) and returns an additional output signal e(n). The filter coefficients of a traditional digital filter do not change over time. The coefficients of an adaptive filter change over time. Therefore,adaptive filters have a self-learning ability that traditional digital filters do not have.Least mean squares (LMS) algorithms are a class of adaptive filter used to mimic a desired filter by finding the filter coefficients that relate to producing the least mean squares of the error signal (difference between the desired and the actual signal). It is stochastic gradient method in that the filter is only adapted based on the error at the current time.

The LMS algorithm is the most popular method for adapting a filter, which have made it widely adopted in many applications. Applications include adaptive channel equalization, adaptive predictive speech coding, Noise Suppression and on-line system identification. Recently, because of the progress of digital signal processors, a variety of selective coefficient update of gradient-based adaptive algorithms could be implemented in practice. The DLMS adaptive algorithm is introduced to achieve lower adaptation-delay. It can be implemented using pipelining. But it can be used only for large order adaptive filters. Typical DSP Programs with highly real-time, design hardware and or software to meet the application speed constraint. It also deals with 3-Dimensional Optimization (Area, Speed, and Power) to achieve required speed, area-power tradeoffs and power consumption. An efficient scheme is presented for implementing the LMS-based transversal adaptive filter in block floating-point

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

(BFP) format, which permits processing of data over a wide dynamic range, at temporal and hardware complexities significantly less than that of a floating-point processor. this work, the performances of adder topologies are tested for robustness against area, delay and power dissipation. They are selected for this work since they have been commonly used in many applications. Addition is an indispensable operation for any high speed digital system,digital signal processing or control system. Therefore pertinent choice of adder topologies is an essential importance in the design of VLSI integrated circuits for high speed and high performance CMOS circuits. The maximum power dissipation occurs for carry save adder. The least power dissipation occurs for ripple carry adder. From the area distribution and gate count the carry save adders occupies more area and gate count, ripple carry occupies less area and Gate count.

3 BLOCK DIAGRAM OF LMS ADAPTIVE FILTER:

There are two main computing blocks in the general block adaptive filter is Error-Computation block and Weight-Update block. The general block diagram of the Delayed LMS adaptive filter is shown in and the delay introduced by the whole of adaptive filters structure. In proposed method the adaptation delay of conventional LMS can be decomposed into two parts: one part of the delay is introduced by the pipeline stages in FIR filtering, and another part of the delay involved in pipelining of weight update process. It can perform optimal pipelining by feed forward cutest retiming of both of these sections separately. Because it is used to minimize the number of pipeline stages and adaptation delay in each of the condition. The Filter output y (n) and the desired signal should be compared and the error signal given to the weight update block.

For every input sample, the LMS algorithm calculates the filter output and finds the difference between the computed output and the desired response. Using this difference the filter weights are updated in every cycle. During the n-th iteration, LMS algorithm updates the weights as follows: Wn+1 = Wn + μ • e(n) • x(n) (1) Where, μ is the convergence-factor. e(n) = d(n) − y(n) y(n)= wTn• x(n) (2) Here, x(n) is the input vector,d(n) is the desired response,and y(n) is the filter output of the nth iteration,w(n) is the weight vector of an Nth order LMS adaptive filter at the nth iteration, respectively, given by

, x(n) = [x(n), x(n − 1), …., x(n − N + 1)]T wn = [wn(0), wn(1),….., wn(N − 1)]T
e(n) Z enotes the error computed in the nth iteration which is used to update the weight.

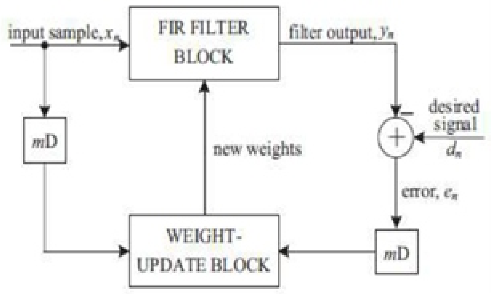
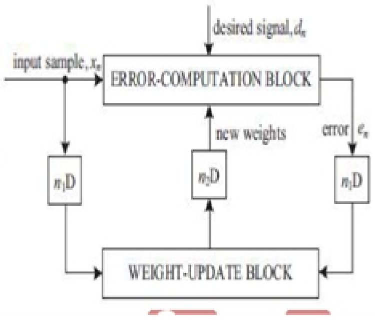


Fig 3.2 (a) Structure of conventional delayed LMS Fig



3.2(b) Structure of modified delayed LMS Adaptive filter Adaptive filter.

Proposed System:

In the conventional DLMS algorithm, the adaptation delay of 'm' cycles amounts to the delay introduced by the whole of adaptive filter structure consisting of FIR filtering and weight adaptation process. But instead, this adaptation delay could be decomposed into two parts. One part is the delay introduced due to the FIR filtering and the other part is due to the delay involved in weight adaptation.

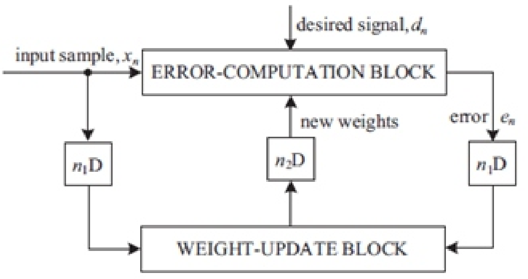


Figure 2: Structure of modified DLMS adaptive filter

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

The proposed adaptive filter architecture, shown in Fig.2, consists of two main computing blocks, namely the error computation block and weight-update block. The computation of filter output and the final subtraction to compute the feedback error are merged in the error computation unit to reduce the latency of error computation path

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.

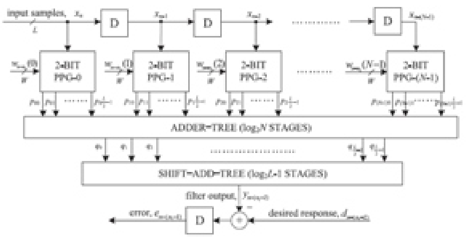


Figure 3: Structure of error-computation block.

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.

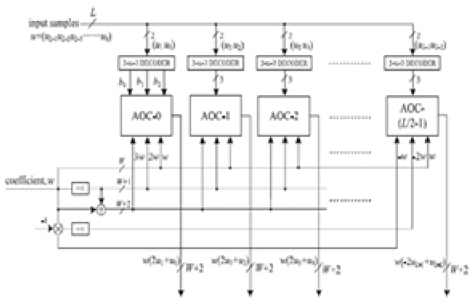


Figure 4: structure of partial product generator

Structure of AOCs:

The structures of an AOC, as shown in Fig 5, 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.

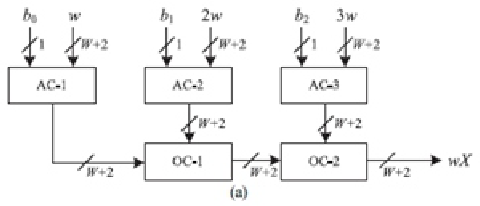


Figure 5: Structure and function of AND/OR cell

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 B. Pipelined Structure Of Weight-Update block The proposed structure of weight-update block is shown in Fig.6. 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 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 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 xi. The sub expression can be shared across all the multipliers. This leads to a gradual reduction of 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.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

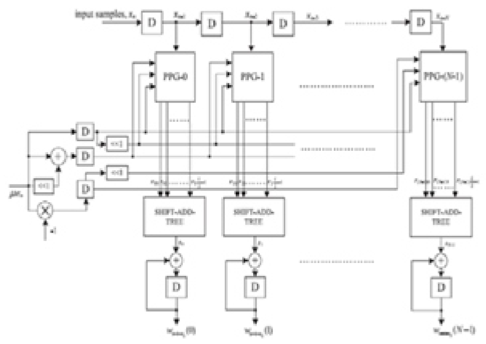


Figure 6: Structure of weight-update block

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 theerror 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.

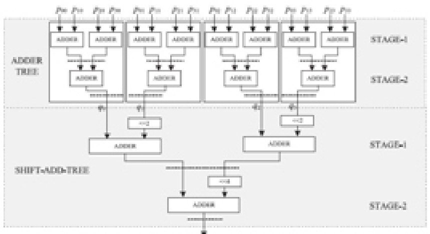


Figure 7: Structure of adder tree

4. SIMULATION RESULTS:

The LMS Filter written in verilog, compiled and simulation using modelsim. The circuit simulated and synthesized. The simulated result for LMS Filter.

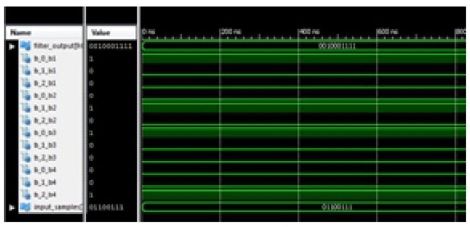


Fig. 7 Simulation Result.

5 CONCLUSION:

We proposed an area–delay-power efficient low adaptation delay architecture for fixed-point implementation of LMS adaptive filter. We used a novel PPG for efficient implementation of general multiplications and inner-product computation by common subexpression sharing. Besides, we have proposed an efficient addition scheme for inner-product computation to reduce the adaptation delay significantly in order to achieve faster convergence performance and to reduce the critical path to support high input-sampling rates. Aside from this, we proposed a strategy for optimized balanced pipeliningacross the time-consuming blocks of the structure to reduce the adaptation delay and power consumption, as well. The proposed structure involved significantly less adaptation delay and provided significant saving of ADP and EDP compared to the existing structures. We proposed a fixed-point implementation of the proposed architecture, and derived the expression for steady-state error. We found that the steady-state MSE obtained from the analytical result matched well with the simulation result.

We also discussed a pruning scheme that provides nearly 20% saving in the ADP and 9% saving in EDP over the proposed structure before pruning, without anoticeable degradation of steady-state error performance. Thehighest sampling rate that could be supported by the ASIC implementation of the proposed design ranged from about 870to 1010 MHz for filter orders 8 to 32. When the adaptive filter is required to be operated at a lower sampling rate, one can use the proposed design with a clock slower than the maximumusable frequency and a lower operating voltage to reduce the power consumption further.

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 pipelined implementation 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 algorithm with 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 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.

[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 Scale Integr. (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, Analog Digital Signal Process., vol. 48, no. 4, pp. 359–366, Apr. 2001.