e-ISSN: 2349-9745
p-ISSN: 2393-8161

Scientific Journal Impact Factor (SJIF): 1.711

International Journal of Modern Trends in Engineering and Research

www.ijmter.com

Area And Power Efficient LMS Adaptive Filter With Low Adaptation Delay

Narmatha.A[1], Nagarajan.p[2]

[1]PG Student, dept of ECE, Vivekanandha college of Engineering for women

[2]Assistant Professor, dept of ECE, Vivekanandha college of Engineering for women

Abstract- We present an efficient architecture for the implementation of delayed least mean square adaptive filter. We use a novel partial product generator and propose a strategy for optimized balanced pipelining across the time consuming combinational blocks of the structure .An efficient systolic architecture of the delayed least mean square adaptive filter based on the processing element .We propose an efficient fixed point implementation of the proposed architecture ,and derive the expression for steady-state error. The architecture is synthesized by using a number of function preserving transformations on the signal flow graph representation of the delayed LMS algorithm.

Key words- Systolic architecture, least mean square (LMS) algorithm, adaptive filters, Fixed-Point LMS Algorithm, LMS Adaptive Filtering

I. INTRODUCTION

Adaptive filters have a wide range of communication and DSP applications. The aim of adaptive filter is to estimate the sequence of scalars from on observation sequence filtered by a system in which coefficients vary. There are two types of adaptive filter. They are recursive least square(RLS),least mean square(LMS).The most widely used algorithm for adaptive filter is least mean square algorithm. The LMS adaptive filter involves a long critical path due to an inner-product computation to obtain the filter output. The critical path is required to be reduced by pipelined implementation when it exceeds the desired sample period. Conventional LMS algorithm does not support the pipelined implementation because of its recursive behavior; it is modified to a form called the delayed LMS algorithm.

II. RELATED WORK

2.1 Systolic architecture

An efficient systolic architecture for the delay least-mean-square (DLMS) adaptive finite impulse response (FIR) digital filter based on a new tree-systolic Processing element (PE) and an optimized tree-level rule. The DLMS algorithm such that the VLSI design of an approximate LMS adaptive finite impulse response (FIR) digital filter could be possible. The new structures that apply the technique, which converts the DLMS algorithm into the LMS algorithm. Both structures approach well the convergence of the LMS algorithm.

2.2 Fixed-Point LMS Algorithm

Adaptive filters with fixed-point arithmetic requires to evaluate the computation quality. The aim of adaptive filters is to estimate a sequence of scalars from an observation sequence filtered by a system in which coefficients vary. These coefficients converge towards the optimum coefficients which minimize the mean quare error (MSE) between the filtered observation signal and the desired sequence. A new model for evaluating the noise power in a fixed-point implementation of the LMS algorithm is presented. This approach has for main advantage to be more tractable than the models to be valid for all types of quantization.

2.3 LMS Adaptive Filtering

An adaptive filter is a system with a linear filter that has a transfer function controlled by variable parameters and a means to adjust those parameters according to an optimized algorithm. An adaptive filter is a computational device that attempts to model the relationship between two signals in real time in an iterative manner. A systolic architecture with minimal adaptation delay and input/output latency, thereby improving the convergence behavior to near that of the original LMS algorithm. The use of delayed coefficient adaptation in the LMS algorithm has enabled the design of modular systolic architectures for real-time transversal adaptive filtering.

2.4 An Efficient Systolic Architecture

The LMS adaptive algorithm minimizes approximately the mean-square error by recursively altering the weight vector at each sampling instance. Thus, an adaptive FIR digital filter driven by the LMS algorithm can be described in vector form as where and denote the desired signal and output signal, respectively. The step-size is used for adaptation of the weight vector, and is the feedback error.

2.5 Least Mean Square

LMS algorithm is used to minimum a desired filter by finding the filter coefficients that relate to producing the least mean squares of the error signal .The LMS algorithms adjust the filter co- efficient to minimize the cost function compared to the RLS algorithms.LMS algorithm do not involve any matrix operations.LMS requires fewer computational resources and many than the RLS algorithm. The implementation of the LMS algorithms also is less complicated than the RLS algorithm . The least mean square (LMS) adaptive filter is the most popular and most widely used adaptive filter, not only because of its simplicity but also because of its satisfactory convergence performance. 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 required to be reduced by pipelined implementation when it exceeds the desired sample period. Since the conventional LMS algorithm does not support pipelined implementation because of its recursive behavior, it is modified to a form called the delayed LMS (DLMS) algorithm ,which allows pipelined implementation of the filter.The DLMS algorithm in systolic architectures to increase the maximum usable frequency .A systolic architecture, where they have used relatively large processing elements (PEs) for achieving a lower adaptation delay with the critical path of one MAC operation.

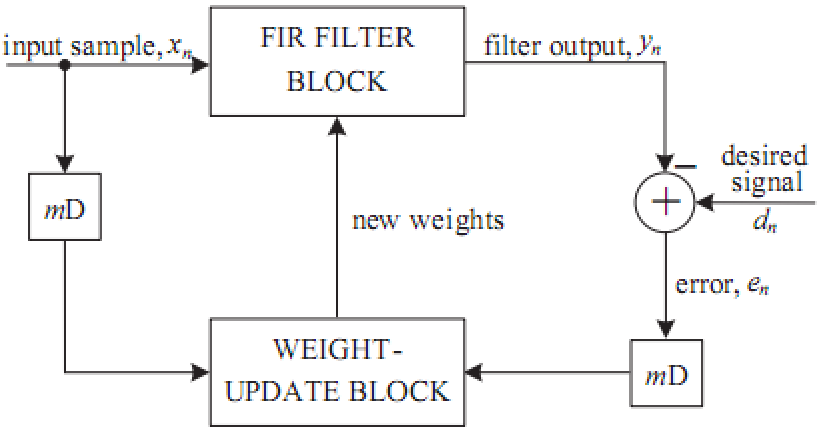

Figure 1. Structure of the conventional delayed LMS adaptive filter.

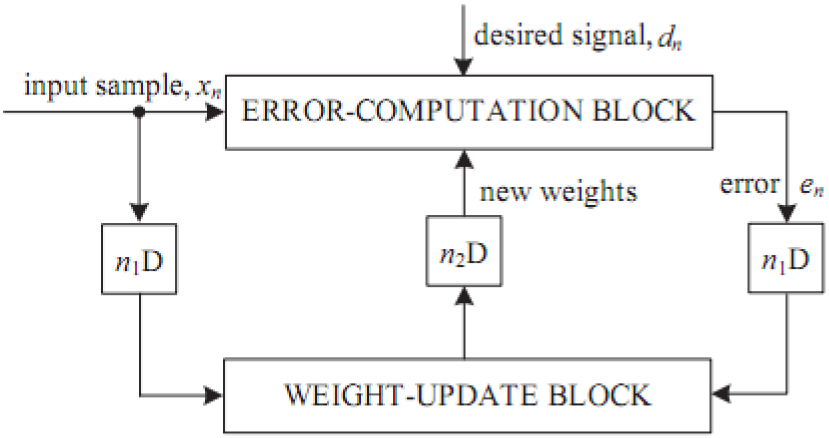

Figure 2. Structure of the modified delayed LMS adaptive filter.

2.6 Delayed LMS Algorithm

The block diagram of the DLMS adaptive filter is shown in Fig. 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. Based on such a decomposition of delay, the DLMS adaptive filter can be implemented by a structure shown in Fig. 2.

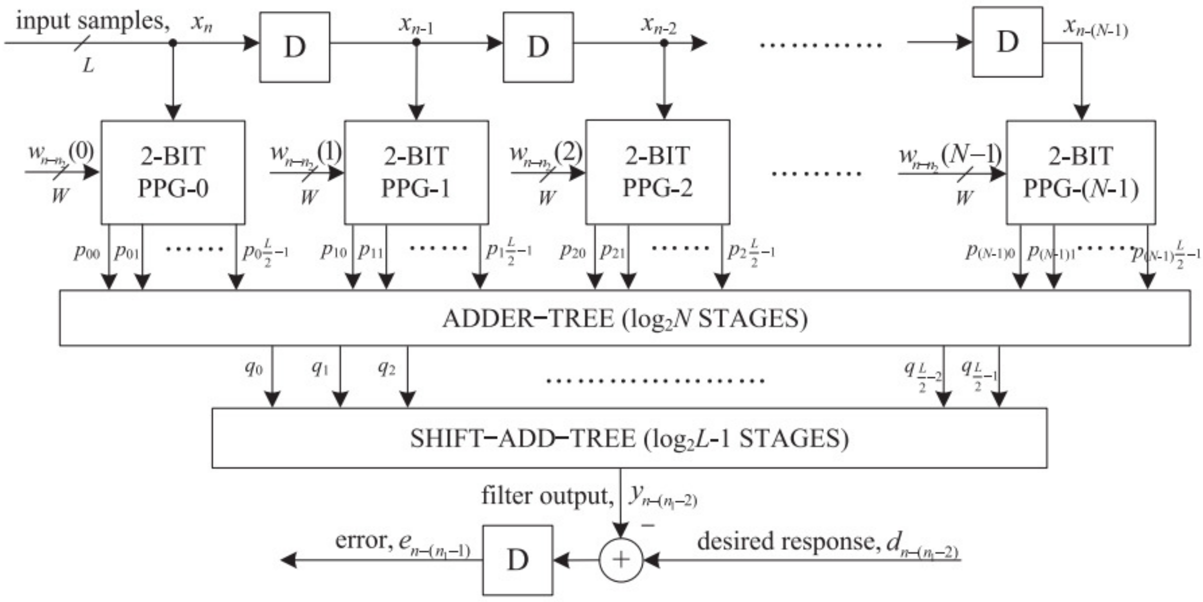
input samples, x_n
L
D
x_{n-1}
x_{n-2}
$x_{n-(N-1)}$
$w_{n-n_2}(0)$
$w_{n-n_2}(1)$
$w_{n-n_2}(2)$
$w_{n-n_2}(N-1)$
W
2-BIT PPG-0
2-BIT PPG-1
2-BIT PPG-2
2-BIT PPG-(N-1)
p_{00} p_{01} $p_{0\frac{L}{2}-1}$
p_{10} p_{11} $p_{1\frac{L}{2}-1}$
p_{20} p_{21} $p_{2\frac{L}{2}-1}$
$p_{(N-1)0}$ $p_{(N-1)1}$ $p_{(N-1)\frac{L}{2}-1}$
ADDER−TREE ($\log_2 N$ STAGES)
q_0 q_1 q_2 $q_{\frac{L}{2}-2}$ $q_{\frac{L}{2}-1}$
SHIFT−ADD−TREE ($\log_2 L$-1 STAGES)
filter output, $y_{n-(n_1-2)}$
error, $e_{n-(n_1-1)}$
desired response, $d_{n-(n_1-2)}$
+
−

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

III. PROPOSED SYSTEM

There are two main computing blocks in the adaptive filter architecture: 1) the error- computation block, and 2) weight-update block. In this Section, we discuss the design strategy of the proposed structure to minimize the adaptation delay in the error-computation block, followed by the weight-update block.

3.1 Partial product generator

A product formed by multiplying the multiplicand by one digit of the multiplier has more than one digit partial products are used as intermediate in calculation. The result obtained when a number is multiplied by one digit of multiplier .The PPG consist of L/2 number of 2- to-3 decoders and the same number of AND/OR cells To take care of the sign of the input samples while computing the partial product corresponding to the most significant digit (MSD).

3.2 Fixed-Point Design Considerations

For fixed -point implementation, the choice of word lengths and radix points for input samples, weights, and internal signals need to be decided. given as the 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. LSBs of weight increment terms are truncated so that the terms have the same fixed-point representation as the weight values. We also assume that no overflow occurs during the addition for the weight update. Otherwise, the word length of the weights should be increased at every iteration, which is not desirable. The assumption is valid since the weight increment terms are small when the weights are converged. MSBs in the computation of the shift–add tree of the weight-update circuit are to be retained, while the rest of the more significant bits of MSBs need to be discarded. This is in accordance with the assumptions that, as the weights converge toward the optimal value, the weight increment terms become smaller, and the MSB end of error term contains more number of zeros.

3.3 Adder-Tree Optimization

The adder tree and shift–add tree for the computation of can be pruned for further optimization of area, delay, and power complexity. To illustrate the proposed pruning optimization of adder tree and shift–add tree for the computation of filter output, we take a simple example of filter length N = 4, considering the word lengths L and W to be 8. The dot diagram contains 10 dots, which represent the partial products generated by the PPG unit, for W = 8. We have four sets of partial products corresponding to four partial products of each multiplier, since L = 8. Each set of partial products of the same weight values contains four terms, since N = 4. The final sum without truncation should be 18 b. However, we useonly 8 b in the final sum, and the rest 10 b are finally discarded.To reduce the computational complexity, some of the LSBs of inputs of the addertree can be truncated, while some guardbits can be used to minimize the impact of truncation on the error performance of the adaptive filter.

IV. 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 sub expression sharing. we proposed a strategy for optimized balanced pipelining across the time-consuming blocks of the structure to reduce the adaptation delay and power consumption, as well. 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.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] R. Rocher, D. Menard, O. Sentieys, and P. Scalart, “Accuracy evaluation of fixed-point LMS algorithm,” in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process., May 2004, pp. 237–240.