Area-Delay-Power Efficient Fixed-Point LMSAdaptive Filter

Kavali Santhosh Yadav[1], N. Praveen kumar[2]
[1]*M.Tech student,ECE,Stanley Stephen College of Engineering and Technology,*
[2]*Associate Professor,ECE ,Stanley Stephen College of Engineering and Technology,*
Kurnool, Andhra Pradesh, India

***Abstract*- This paper presents the modified delayed LMS adaptive filter consists of Weight update block with Partial Product Generator (PPG) to achieve a lower adaptation delay and efficient area, power, delay. To achieve lower adaptation delay, initially the transpose form LMS adaptive filter is designed but the output contains large delay due toits inner product process. Here, the pipelining structure isproposed across the time consuming combinational blocks of the structure to reduce the critical path.**

***Index Terms*- Partial Product Generator, Weight updateblock, Modified DLMS adaptive filter.**

I. INTRODUCTION

The Least Mean Square (LMS) adaptive filter is the widely used filter because of its simplicity and performance. 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 Least Mean Square adaptive filter is used here because it differs from a traditional digital filter in the following 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.

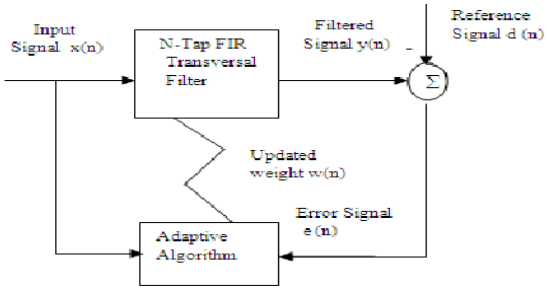

Fig.1: Adaptive filter system

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.

However the implementation of constant multiplications in a shift-adds architecture enables the sharing of common partial products among the constant multiplications that significantly reduces the area and power dissipation of the MCM design. Hence, the MCM problem is defined as finding the minimum number of addition/subtraction operations that implement the constant multiplications, since shifts can be realized using only wires in hardware. Note that the MCM problem is an NP-complete problem [4]. Most work on implementation of digit-serial FIR filters has focused on implementation in FPGAs and without using multiplier blocks [11]–[13]. However in [14] the digit-size trade-off in implementation of digit-serial transposed direct form FIR filters using direct multiplier blocks was studied.

II. RELATED WORKS

This algorithm is a class of adaptive filter used to mimic adesired filter by finding the filter

coefficients that relate to producing the least mean squares of the error signal [1]. TheLMS algorithm was devised for the study of a pattern recognition machine known as the adaptive linear element.The LMS algorithm is a stochastic gradient algorithm in that ititerates each tap weight of the transversal filter in the direction of the instantaneous gradient of the squared error signal withrespect to the tap weights [2]. The existing systolic architectures for the LMS algorithm with delayed coefficient adaptation have large adaptation delay and hence degraded convergence behaviour. The proposed system gives the systolic architecture with minimal adaptation delay andinput/output latency, there by improving the convergence behaviour to near that of the original LMS algorithm. [3]. An efficient systolic architecture for the DLMS adaptive filter is based on a new tree-systolic processing element (PE) and an optimized tree-level rule. Applying tree-systolic, a higher convergence rate than that of the conventional DLMS structures can be obtained without the properties of the systolic-array architecture [4].

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 [5]. 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 [6].An efficient scheme is presented for implementing the LMS-based transversal adaptive filter in block floating-point (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 [7]. The implementation of adaptive filters with fixed-point arithmetic requires to evaluate the computation quality. The accuracy may be determined by calculating the global quantization noise power in the system output [8]. The LMS algorithm is the most popular method for adapting a filter, which is used in many applications such as adaptive channel equalization,adaptive predictive speech coding, Noise Suppression and online system identification.

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.

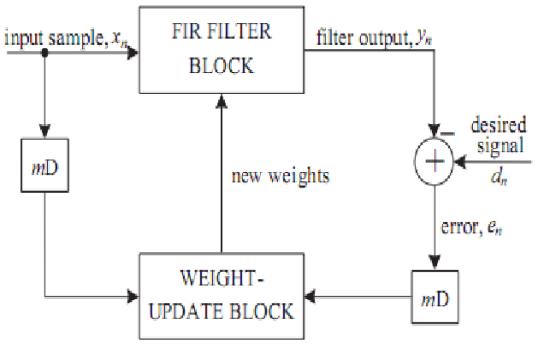

Figure 1. Structure of the conventional delayed LMS adaptive filter

Based on such decomposition of delay, the proposed structure of DLMS adaptive filter is shown in Fig.3.

III. PROPOSED SYSTEM

The proposed adaptive filter architecture 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. If the latency of computation of error is n1 cycles, the error computed by the structure at the n^{th} cycle is e(n − n1), which is used with the input samples delayed by n1 cycles to generate the weight-increment term. The weight update equation of the proposed delayed LMS algorithm is, therefore, given by,

$w_{n+1} = w_n + \mu \cdot e(n - n1) \cdot x(n - n1)$……. (3a)

Where,

$e(n - n1) = d(n - n1) - y(n - n1)$ ……(3b)

and

$y(n) = w_{Tn-n2} \cdot x(n)$……. (3c)

We can notice that during weight adaptation, the error with n1 delays is used while the filtering unit uses the weights delayed by n2 cycles. By this approach the adaptation-delay is effectively reduced by n2 cycles. The proposed algorithm canbe implemented efficiently with very low adaptation-delay which is not effected substantially by the increase in filter order.

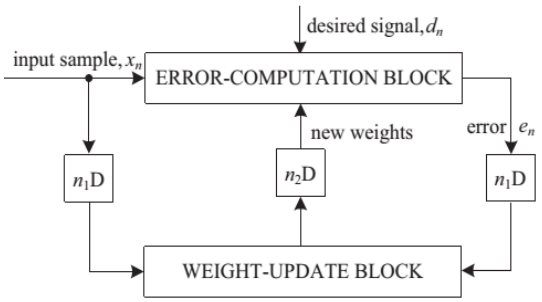

Fig. 2.Structure of the modified delayed LMS adaptive filter.

A. Error-Computation Block

The proposed structure for error-computation unit of an Ntap 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 subblock is described in detail.

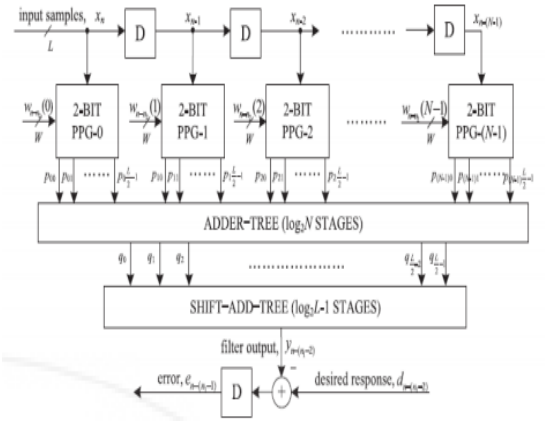

Figure 3: Structure of error-computation block.

B. Multiple Constant Multiplication: The Multiplication of a variable by a set of constants, generally known as Multiple Constant Multiplications (MCM).It is essential in many Digital Signal Processing (DSP) applications,

- Digital Finite Impulse Response (FIR) filters,
- Fast Fourier Transforms (FFT) and
- Discrete Cosine Transforms (DCT)**.**

C. Wallace Tree Multiplier: A Wallace tree is an efficient hardware implementation of a digital circuit that multiplies two integers. The Wallace tree has three steps:

a) Multiply (that is - AND) each bit of one of thearguments, by each bit of the other, yielding results.

Depending on position of the multiplied bits, the wirescarry different weights.

b) Reduce the number of partial products to two by layers of full and half adders

c) Group the wires in two numbers, and add them with a conventional adder.

A typical Wallace tree architecture is shown in figure4 below. In the diagram AB0-AB7 represents the partial products the partial products.

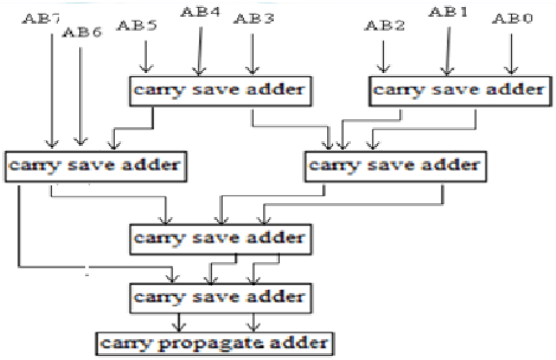

Figure 4: Wallace tree multiplier

D. Partial product generator

A product formed by multiplying the multiplic and by one digit of the multiplier has more than onem 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 thesame number of AND/OR cells To take care of the sign of the input samples while computing th partial product corresponding to the most significant digit (MSD).

E. 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 specificscaling/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.

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.

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 ofdelayed 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-LMSfilters," J. Very Large Scale Integr. (VLSI) Signal Process., vol. 39, nos. 1–2, pp. 113–131, Jan. 2005.
[6] H. Herzberg and R. Haimi-Cohen, "A systolic array realization of anLMS 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 SignalProcess., 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 LargeScale 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 forthe DLMS adaptive filter and its applications," IEEE Trans. CircuitsSyst. II, Analog Digital Signal Process., vol. 48, no. 4, pp. 359–366,Apr. 2001.
[11] L.-K. Ting, R. Woods, and C. F. N. Cowan, "Virtex FPGA implementation of a pipelined adaptive LMS predictor for electronic supportmeasures 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 LMS adaptive filter part-I: Introducing a novel multiplication cell," 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 filterpart-II: An optimized architecture," in Proc. IEEE Int. Midwest Symp.Circuits Syst., Aug. 2011, pp. 1–4.
[15] K. K. Parhi, VLSI Digital Signal Procesing Systems: Design and Implementation. New York, USA: Wiley, 1999.

BIODATA

Author

Kavali Santhosh Yadav presently pursuing his M.Tech in VLSI System Design from Stanley Stephen College of Engineering and Technology, Kurnool, Andhra Pradesh, India.

CoAuthor

N.Praveen kumar has received B.Tech Degree (Electronics & Communication Engineering) in the year 2007, M.Tech (VLSID) in 2009 and also pursuing Ph.D. (VLSI) from JNTU Anantapur, A.P, India. Presently he is working as Associate Professor in the Dept. of ECE, SSCET, Panchalingala, Kurnool, India.