INTERNATIONAL JOURNAL OF RESEARCH
In Advanced Engineering Technologies

Volume 5, Issue 1 SEP 2015

AREA-DELAY-POWER EFFICIENT FIXED-POINT LMS ADAPTIVE FILTER WITH LOW ADAPTION-DELAY

[1]**CHAND BAIG M**, PG Scholar In VLSI System Design,
[2]**CHENNAIAHM.P**,Associate. Professor, ECE Department,
SRISAI INSTITUTE OF SCIENCE AND TECHNOLOGY

Abstract: In this paper, the fir filter isproposed an efficient multiplication stage technique. To investigate the area, speed, power tradeoffs for implementation of FIR filters using MCM and digit-serial arithmetic. Multipleconstant multiplications (MCM) are an efficient way of implementing several constant multiplications with the same input data. The coefficients of multiplier are expressed using shifts, adders, and subtractions. To introduce the Wallace tree multiplier for reduce both the number ofadders and subtractions as well as the number of shifts. This method can be used to reduce the amount of embedded multipliers in large MCM blocks. We use Xilinx 14.5 toprovide VHDL coding for our architecture. Result shows the better performance rate of our proposed work than existing algorithms. This paper is used to reduce the delay product and reduce the area and power consumption in FIR filtering.

Keywords: Multiple constant multiplication, Wallace tree architecture, RPAGalgorithm, FIR filter, Pipelining, Partial product generation,Adder depth

I. INTRODUCTION

The Least Mean Square (LMS) adaptive filter is the widely used filter because of its simplicity and performance. Least mean squares (LMS) algorithm ms are a class of adaptive filterused 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 algorith m is the most popular method for adapting a filter, which have made it widely adopted in many application s. 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 dig ital 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. The filter is an important component in the communication world. It can eliminate unwanted signals fro m 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.

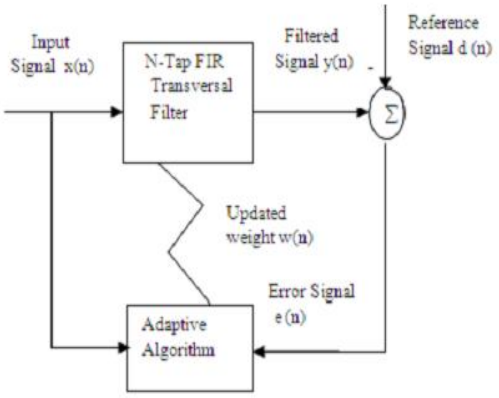

Fig.1: Adaptive filter system

II. EXISTING SYSTEM

This algorithm is 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 [1]. The LMS algorith m was devised for the study of a pattern recognition machine known as the adaptive linear element. The LMS algorith m is a stochastic gradient algorithm in that it iterates each tap weight of the transversal filter in the direction of the instantaneous gradient of the squared error signal with respect to the tap weight s [2]. The existing systolic architectures for the LMS algorith m with delayed coefficient adaptation have large adaptation delay and hence degraded convergence behaviour. The proposed system gives the systolic arch itecture with minimal adaptation delay and input/output latency, thereby improving the convergence behaviour to near that of the original LMS algorith m. [3]. Anefficient 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 arch itecture [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 co mplexities significantly less than that of a floating-point processor [7]. Theimplementation 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 on line system identification.

In existing system, they proposed fixed point adaptive filter with low adaptive delay for optimized balanced pipelining across the time-consuming combinational blocks of the structure. 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. Drawbackof the existing is to slow convergence (due to Eigen-value spread).This length of time might cause problems in these applications because the adaptive filter must work in real time to filter the input signals. The steady-state behavior of the LMS algorithm, the step size is small. The DLMS algorithm is reduced the convergence speed and poorer tracking performance. It increases output latency.

Figure 2: Structure of Delayed LMS Adaptive Filter

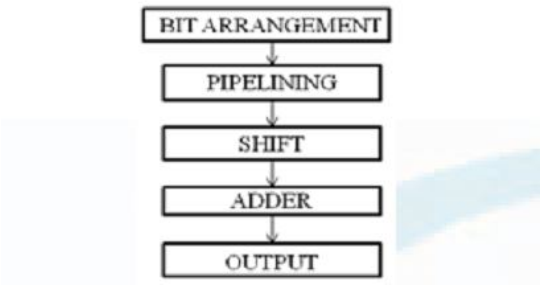

Figure 3: Block Diagram of Existing System

III. DELAYED LMS ADAPTIVE FILTER

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 inevery cycle. During the n-th iteration, LMS algorithm updates the weights as follows:

$$W_{n+1} = w_n + \mu \cdot e(n) \cdot x(n) \quad (1a)$$

Where,

$$e(n) = d(n) - y(n)$$
$$y(n) = wTn \cdot x(n) \quad (1b)$$

Here,

x(n) is the input vector
w(n) is the weight vector of an Nth order LMS adaptive filter
at the nth iteration, respectively, given by ,

$$x(n) = [x(n), x(n-1), \cdots, x(n-N+1)]^T$$
$$w_n = [w_n(0), w_n(1), \cdots, w_n(N-1)]^T$$

d(n) is the desired response and y(n) is the filter output of the nth iteration.e(n) denotes the error co mputed in the nthiteration which is used to update the weights.μ is the convergence-factor.

The DLMS algorithm, instead of using the recent -most feedback-error e(n) corresponding to the n-th iteration for updating the filter weights, it uses the delayed error e(n−m),(i.e.) the error corresponding to (n−m)-th iteration for updatingthe current weight. The weight -update equation of DLMS algorithm is given by,

$$W_{n+1} = w_n + \mu \cdot e(n-m) \cdot x(n-m) \quad (2)$$

Where, m is the adaptation-delay.

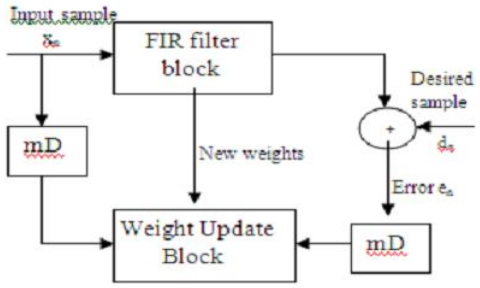

Fig 4: Structure of delayed LMS adaptive filter

The structure of conventional delayed LMS adaptive filter is shown in Fig. 1. It can be seen that the adaptation-delay m is the number of cycles required for the error corresponding to any given sampling instant to become available to the weight adaptation circuit.

IV.PROPOSED SYSTEM

In the conventional DLMS algorithm (Fig.1) 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.

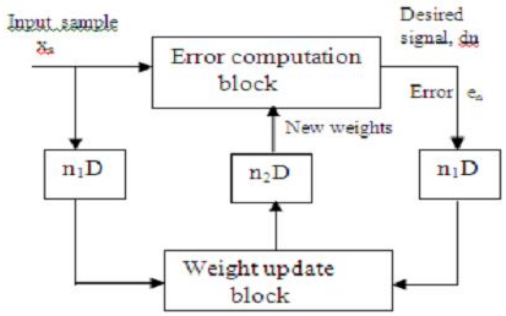

Fig 5: Structure of modified DLMS adaptive filter

Based on such decomposition of delay, the proposed structure of DLMS adaptive filter is shown in Fig.5. The proposedadaptive 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 n1cycles, the error computed by the structure at the nth 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 ,

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

Where,

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

And

$$y(n) = wTn{-}n2 \cdot x(n) \quad (3c)$$

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 can be implemented efficiently with very low adaptation -delay which is not effected substantially by the increase in filter order

Error computation block

The error computation block is implemented by a pipelinedinner-product computation module (Fig.5) and the weight update block is implemented by N number of pipelinedmultiply-accumulate units. The multiplications of input samples with corresponding weights followed by their summation in the filter computation block have been realized by a carry-save chain and the error computation is merged with the filter output computation. In the weight update block, the multiplication of the convergence -factor with the error and input values are combined with the addition with old weights according to (3a) to obtain the final updated weights in N parallel carry-save chains.

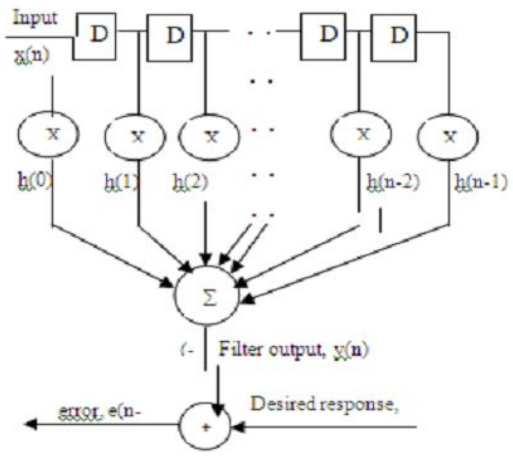

Fig 6: Structure of Error-co mputation block

Weight update block

The function of weight-update block is shown in Fig.6. Theconvergence-factor is taken to be a negative power of two to realize the corresponding multiplication of (3a) by a shiftoperation. The weight-update block consists of N carry –saveunits to update N weights. Each of those carry -save units performs the multiplication of shifted error values with thedelayed input samples along with the addition with the oldweights. Note that the addition of old weight with weightincrement term is merged with multiplication pertaining to the calculation of weight -increment term.

The final outputs of the carry -save units constitute the updatedweights which serve as an input to the error computation blockas well as the weight-update block for the next iteration. A pair of delays are introduced before and after the final addition of the weight -update block to keep the critical-path equal to one addition time. The shaded region in Fig .6 indicates the carry- save unit corresponding to the first weight which takes the delayed input samples and shifted from of delayed error value (to take care of the multiplication by convergence factor) and the old weights as input. Thus the weight -update block takes a total of two delays, i.e., n2 = 2.

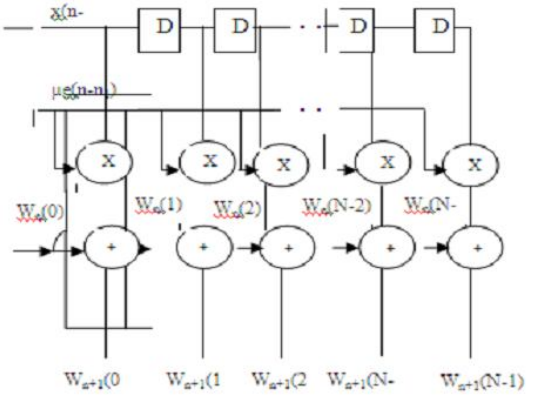

Fig 7: Structure of Weight update block

V.PERFOMANCE RESULTS

This section evaluates the performance of the proposed modified least mean square (LMS) algorithm and shows the simulation results. The first result declares

about the output of LMS adaptive filter with delay. It is having some delay in the output of Least Mean Square adaptive filter.And the second result declares about the output of LMS adaptive filter withoutdelay. After the clock input has given the output of the adaptive filter is achieved without delay. The ModelSIMis the tool used here to check the performance of LMS adaptive filter. It is a complete HDL simulation environment thatenables to verify the source code and functional and timing models using test bench .

Output of LMS Adaptive Filter With Delay

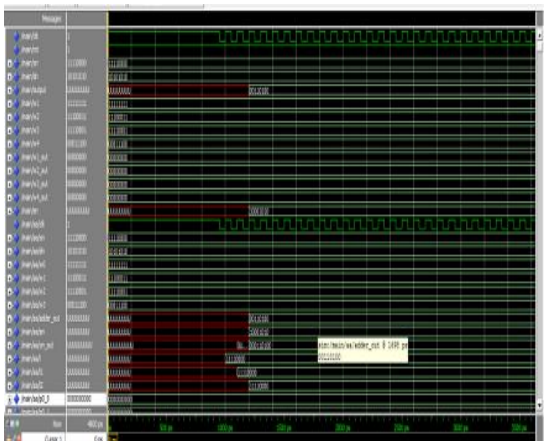

Fig 8: Output with delay

Output of LMS Adaptive Filter without Delay

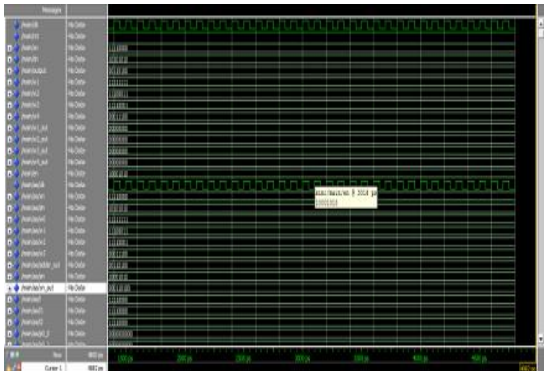

Fig 9: Output without delay

CONCLUSION

In this paper implementation of low power digit-serial FIR filters using multiple constant multiplication (MCM) techniques has been considered. Some conclusions regarding design guidelines for low power digit-serial multiplier blocks can be deduced. The actual complexity in terms of adder cost and number of shifts is not the main factor determining the power consumption. Instead the adder depth, as for parallel arithmetic, is a main contributor. Hence, an algorithm with low adder depth should be used. Furthermore, the shifts prevent glitch propagation through subsequent adders. For even coefficients the shifts can be placed either before or after the final additions. Hence, a heuristic for placing the shifts would be also useful. Better results were achieved using RPAG compared to optimally pipelined adder graphs for FPGAs and the method presented in targeting ASICs. Furthermore, it was shown that the algorithm often finds better solutions for minimal total AD compared to existing system.

REFERENCES

[1] Y. Yi, R. Woods, L.-K.Ting, and C. F. N. Cowan, "High speed FPGA-based implementations of delayedLMS filter," J. Very Large Scale Integr. (VLSI) Signal Process, vol. 39, nos. 1–2, pp. 113–131, Jan. 2005.

[2] L.-K. Ting, R. Woods, and C. F. N. Cowan, "Virtex FPGA implementation of a pipelined adaptive LMS predictor for electronic support measures receivers," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 13, no. 1, pp. 86–99, Jan. 2005

[3] P. K. Meher and M. Maheshwari, "A high-speed FIR adaptive filter architecture using a 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.

[4] 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.

[5] 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.

[6] FIRsuite, "Suite of constant coefficient FIRfilters,"2011.[Online].Available:http://www.firsuie.net

[7] K. Parhi, VLSI Digital Signal Processing Systems: Design and Implementation. John Wiley & Sons, 1999.
[8] G. Dempster and M. D. Macleod, "Use of minimumadder multiplier blocks in FIR digital filters," IEEE Trans. Circuits Syst. II, AnalogDigit.Signal Process., vol. 42, no. 9, pp. 569–577, Sep. 1995.
[9] M. Kumm and P. Zipf, "High speed low complexity FPGA-based FIR filters using pipelined adder graphs," in Field Programmable Technology, International Conference on (ICFPT), 2011
[10]L. Aksoy, E. Costa, P. Flores, and J. Monteiro, "Optimization of gatelevel area in high throughput multiple constant multiplications," in Proc. European Conf. on Circuit Theory and Design.
[11]S. Mirzaei, R. Kastner, and A. Hosangadi, "Layout aware optimization of high speed fixed coefficient FIR filters for FPGAs," International Journal of Reconfigurable Computing, vol. 3, pp. 1 – 17, January 2010
[12]U. Meyer-Baese, J. Chen, C. H. Chang, and A. G. Dempster, "A comparison of pipelined RAG-n and DA FPGA-based multiplierless filters," Circuits and Systems, 2006. APCCAS 2006. IEEE Asia Pacific Conference on, pp. .1555–1558, 2006.
[13]Voronenko, "Spiral website," 2011.[Online]. Available: http://www.spiral.net/hardware/multless.html
[14]M. Faust and C. H. Chang, "Minimal logic depth adder tree optimization for multiple constant multiplication," in Proc. IEEE Int. Symp. On Circuits Syst., 2010. ISCAS 2010, Paris, France, May 30 - Jun. 2 2010,pp. 457–460.
[15]L. Aksoy, E. Costa, P. Flores, and J. Monteiro, "Optimization of area and delay at gate-level in multiple constant multiplications," in Digital System Design: Architectures, Methods and Tools (DSD), 2010 13thEuromicro Conference on, September 2010, pp. 3–10.
[16]K. Johansson, "Low power and low complexity shiftand-add based computations," Ph.D. dissertation, Link¨oping University, 2008, Link¨oping Studies in Science and Technology. Dissertations
[17]Y. Voronenko and M. P¨uschel, "Multiplierless multiple constant multiplication," ACM Trans. Algorithms, vol. 3, no. 2, p. 11, May 2007.

INTERNATIONAL JOURNAL OF RESEARCH
In Advanced Engineering Technologies

Volume 5, Issue 1 SEP 2015