LUT Optimization of Adaptive Filter using Distributed Arithmetic

Rajesh. R *, N. Porutchelvam
Department of Electronics and Communication Engineering, Gnanamani College of Technology, Namakkal, Tamil Nadu, India

Article Info
Article history:
Received 3 February 2015
Received in revised form
20 February 2015
Accepted 28 February 2015
Available online 6 March 2015

Keywords
Adaptive Filter,
Distributed Arithmetic,
System Verilog

Abstract
Distributed arithmetic (DA) approach is preferable only for implementing the DSP applications in FPGAs. These DSP applications always prefer the efficient memory based computation for enhanced functionality. On considering an efficient computation, a LUT based memory array is designed. Further, the conventional method focuses only on identifying the potential element such as area, power and throughput, but fails to produce the optimized results. In this paper describes the combined effort of APS-OMS technique. This technique reduces the LUT size by the factor of four than the existing one. Furthermore, it optimizes the area and power consumption. The adaptive filter designs are simulated by using the Modelsim and synthesis using Xilinx.

1. Introduction

Adaptive filters is an another important applications of signal processing area including acoustic echo cancellation, signal denoising, sonar signal processing, clutter rejection in radars, and channel equalization for communication and networking systems. The famous Haykin and Widrow least mean square (LMS) are most commonly used adaptive filter for weight updated and filtering. It does not support high sampling period but due to suitable convergence time [1]. The direct form of FIR filter configuration has long critical path due to an inner product computation to get a filter output. Multiplier less distributed arithmetic (DA) based technique [2] has gained substantial popularity for its high-throughput processing capability and regularity, which result in cost effective and area–time efficient computing structures. Hardware-efficient DA-based design of adaptive filter has been suggested by Allred et al. [3] using two separate lookup tables (LUTs) for filtering and weight update. Guo and DeBrunner [4], [5] have improved the design in [3] by using only one LUT for filtering as well as weight updating. However, the structures in [3]–[5] do cycles for LUT updates for each new sample. In previous paper, DA-based architectures has low-area, low-power and high throughput of adaptive filter with very high area delay product [6]. In this paper analyses and describes the optimization of LUTs using adaptive filter. Conventional LUTs require memory for all combination of inputs. The combination of input subjected to proceed based on OMS that utilizes the APC to reduce the LUT size to one-fourth of its conventional size. The signed and unsigned operands both are used in APC-OMS technique. Area complexity can be reduces by the higher order inputs of LUTs. The rest of this brief is organized below in sub sections.

2. Lms Adaptive Algorithm

During each cycle, the LMS algorithm computes a filter output and an error value that is equal to the difference between the current filter output and the desired response. The estimated error is then used to update the filter weights in every training cycle. The weights of LMS adaptive filter during the nth iteration are updated according to the following equations:

$W(n+1) = w(n) + \mu.e(n).y(n)$

Where,

$E(n) = d(n)-y(n)$

$Y(n) = w^{qt}(n)\cdot x(n)$

The input vector x(n) and the weight vector w(n) at the nth training iteration are respectively given by

$X(n)=[x(n), x(n-1),...,x(n- N + 1)]^T$

$W(n)=[w0(n), w1(n),...,wn-1(n)]T$

D(n) is the desired response, and y(n) is the filter output of the nth iteration. E(n) denotes the error computed during the nth iteration, which is used to update the weights, μ is the convergence factor, and N is the filter length. In the case of pipelined designs, the feedback error e(n) becomes available after certain number of cycles, called the "adaptation delay. The pipelined architectures therefore use the delayed error e(n− m) for updating the current weight instead of the most recent error, where m is the adaptation delay. The weight-update equation of such delayed LMS adaptive filter is given by

$W(n+ 1) = w(n) +\mu. E(n- m).x(n- m)$

3. Conventional Architecture

The conventional DA based adaptive filter structure of length N=4 is shown in Fig.1.It consists of 4-point inner product block, weight increment block, control word generator and sign and magnitude comparator. The four-point inner-product block consists of DA table, multiplexor ad carry save accumulator. DA table consists of an array of 16 registers which stores the partial inner products y_l for $0 < L \leq 15$ and a 16: 1 multiplexor (MUX) to select the content of one of those registers. Bit slices of weights $A = \{w_{3l}\ w_{21}\ w_{11}\ w_{01}\}$ for $0 \leq l \leq L-1$ are fed to the MUX as control in LSB-to-MSB order, and the output of the MUX is fed to the carry-save accumulator After L bit cycles, the carry-save accumulator shift accumulates all the partial inner products and generates a sum word and a carry word of size (L+ 2) bit each. The carry and sum words are shifted added with an input carry"1" to generate filter output which is subsequently subtracted from the desired output d (n) to obtain the error e(n).As in all the bits of the error except the most significant one are ignored, such that multiplication of

Corresponding Author,
E-mail address: rajesh92b@gmail.com

input x_k by the error is implemented by a right shift through the number of locations given by the number of leading zeros in the magnitude of the error.

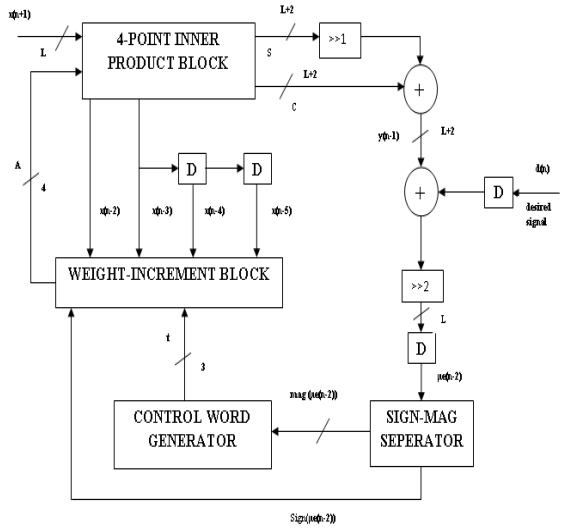

Fig: 1. DA based LMS Adaptive Filter

The magnitude of the computed error is decoded to generate the control word t for the barrel shifter. The logic used for the generation of control word t to be used for the barrel shifter. The convergence factor μ is usually taken to be O (1/N). We have taken μ =1/N. However, one can take μ as 2^{-I} /N, where I is a small integer. The number of shifts t in that case is increased by i, and the input to the barrel shifters is preshifted by i locations accordingly to reduce the hardware complexity. The weight-increment unit for N = 4 consists of four barrel shifters and four add/sub cells. The barrel shifter shifts the different input values X_k for k =0, 1... N −1 determined by the location of the most significant one in the estimated error. The barrel shifter yields the desired increments to be added with or subtracted from the current weights. The sign bit of the error is used as the control for add/sub cells such that, when sign bit is zero or one, the barrel-shifter output is respectively added with or subtracted from the content of the corresponding current value in the weight register. In conventional architecture have high power, high throughput and high area when compared to the Distributed arithmetic.

4. Proposed Architecture

In the proposed DA based adaptive filter structure consists of two blocks filter convolution using MUX and coefficients updating using APS-OMS technique.

The combined APS based OMS technique is used to reduce the LUTs size. It consists of address generator, LUT address, barrel shifter and LUT output. Initially, the binary input is given to the address which converts the input to match the address of the LUT. The matched LUT address are thus stored on to the barrel shifter and finally the relevant output is generated with the help of add/sub unit. The combined APC–OMS technique of the LUT size of adaptive filter input is *L* = 5 as shown in Fig.5. It consists of an LUT of nine words of (W+4)-bit width, a four-to-nine-line address decoder, a barrel shifter, an address generation circuit, and a control circuit for generating the RESET signal and control word (s1s0) for the barrel shifter.

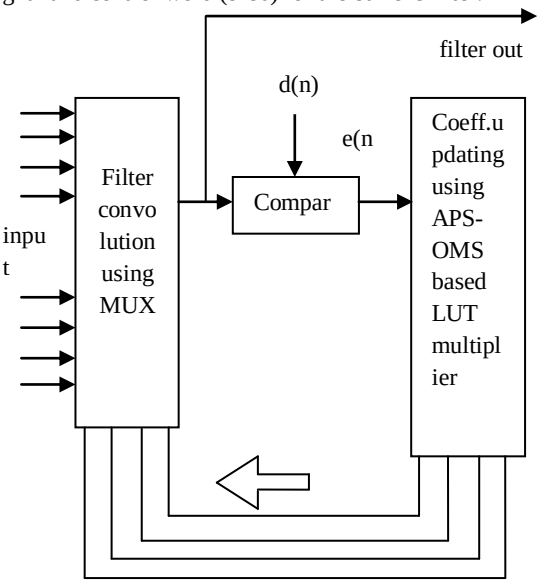

Fig: 2. Proposed DA based LMS Adaptive Filter using APC-OMS Technique

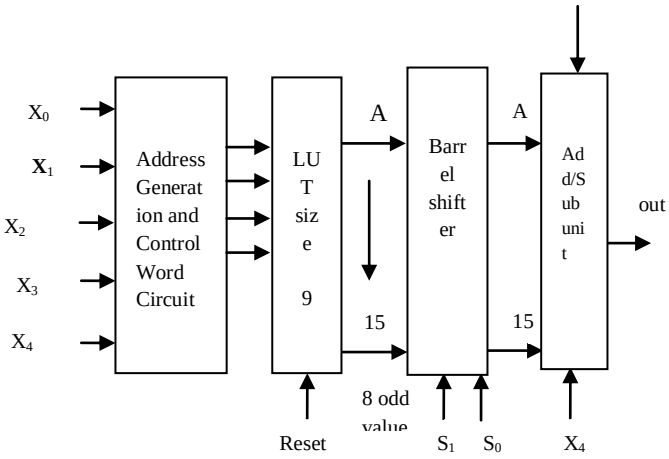

Fig: 3. APS-OMS based LUT Multiplier

The precompiled values of A × (2i + 1) are stored as .P_i for i= 0, 1, 2.....7 at the eight consecutive locations of the memory array, whereas 2Ais stored for input X= (00000) at LUT address "1000," as specified in Table 1. In the above table, is specified as the eight odd multiples, A × (2i + 1) are stored as P_i, for i= 0, 1, 2....7 of eight locations. The even multiples 2A, 4A and 8Aare derived by left-shift operations of A. In the same way 6Aand 12Aare obtained by left shifting 3A, whereas 10Aand 14Aare obtained by left shifting 5A and 7A. A barrel shifter for producing a maximum of three left shifts could be used to derive all the even multiples of A. The word to be stored for X = (0000) is not 0, it is believed to be 16A, which can be derived from Aby four left shifts using a barrel shifter. On the other hand, if 16Ais not derived from A, only a maximum of three left shifts is necessary to achieve all other even multiples of A.

Table: 1. OMS based LUT storage APS Technique

Address	Stored Value
0000	1A
0001	3A
0010	5A
0011	7A
0100	9A
0101	11A
0110	13A
0111	15A
1000	2A

A maximum of three bit shifts can be implemented by a two-stage logarithmic barrel shifter, but the implementation of four shifts requires a three-stage barrel shifter. As a result 2Aat (1000), the product 16A can be obtained by three arithmetic left shifts which are necessary to locate X = (00000). For X= (00000), the desired encoded word 16Ais obtained by 3-bit left shifts of 2Awhich is stored at address (1000).

5. Results and Discussion

The design is coded using verilog and simulated using Modelsim and implemented using FPGA. The given input signal is a noise signal and the desired signal as the original signal and the error signal is filtered out and output is generated as original signal.

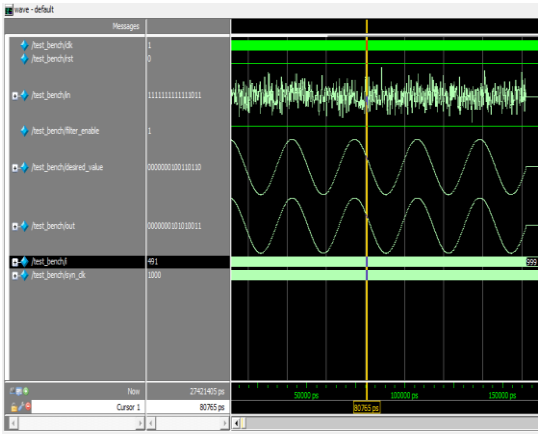

Fig: 4. Simulation Result for DA based LMS adaptive filter

Table: 2. Simulation Parameters

Design	Area(sq.µm)	Power(mW)
Conventional	9431	91
Proposed	9236	81

6. Conclusion

In this paper, a combined APS-OMS technique was proposed for the adaptive filter. The combined APS-OMS technique reduces the LUT size of the given inputs. In proposed design have less power, low area and high throughput than its conventional design. The comparison showed that the proposed design has wide range and suitable in the application of digital processing systems. DA based LMS Adaptive filter is well suited for the communication systems.

References

[1] S. Haykin, B. Widrow, Least-Mean-Square Adaptive Filters. Hoboken, NJ, USA: Wiley, 2003

[2] S. A. White, Applications of the distributed arithmetic to digital signal processing: A tutorial review, IEEE ASSP Mag., 6(3), 4–19, 1989

[3] D. J. Allred, H. Yoo, V. Krishnan, W. Huang, D. V. Anderson, LMS adaptive filters using distributed arithmetic for high throughput, IEEE Trans. Circuits Syst. I, Reg. Papers, 52(7), 2005, 1327–1337

[4] R. Guo, L. S. DeBrunner, Two high-performance adaptive filter implementation schemes using distributed arithmetic," IEEE Trans. Circuits Syst. II, Exp. Briefs, 58(9), 2011, 600–604

[5] R. Guo, L. S. DeBrunner, A novel adaptive filter implementation scheme using distributed arithmetic, in Proc. Asilomar Conf. Signals, Syst., Comput., 2011, 160–164