INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH & DEVELOPMENT

ISSN 2278 – 0211 (Online)

Area and Power Efficient Fixed Point LMS Adaptive Filter using Ripple Carry Adder

Rubiya C. H.
Department of Electronics and Communication Engineering
Christ the King Engineering College, Coimbatore Tamil Nadu, India
Muralidharan V.
Department of Electronics and Communication Engineering
Christ the King Engineering College, Coimbatore Tamil Nadu, India
Varatharaj M.
Department of Electronics and Communication Engineering
Christ the King Engineering College, Coimbatore Tamil Nadu, India

Abstract
The area and power efficiency is very important for every circuit and its applications. Here we we present an efficient architecture for the implementation of a delayed least mean square adaptive filter. By using improved adder structure the area power efficiency can be increased. with this the area delay product (ADP) and energy delay product(EDP) can be saved in a considerable amount. Ripple carry adder is used here and the main advantages are lower power consumption as well as compact layout giving smaller chip area..

Key words*: Adders, Adaptive filter, LMS algorithm, fixed point arithmetic*

1. Introduction

THE LEAST MEAN SQUARE (LMS) adaptive filter is the most common adaptive filter, not only because of its simplicity but also because of its satisfactory convergence performance. The aim of adaptive filters is to estimate a sequence of scalars from an observation sequence filtered by a system in which coefficients vary. In the Conventional LMS adaptive filter, the estimated signal in each data interval is computed and subtracted from the desired signal. The error is then used to update the top coefficients before the next sample arrives. In some practical applications the LMS adaptive scheme imposes a critical limit on its implementations.

The existing work on the DLMS adaptive filter does not discuss the fixed-point implementation issues, e.g., location of radix point, choice of word length, and quantization at various stages of computation, although they directly affect the convergence performance, particularly due to the recursive behavior of the LMS algorithm. Therefore, fixed-point implementation issues are given adequate emphasis in this paper. Besides, we present here the optimization of our previously reported 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.

To give high accuracy in area and power the ripple carry adder structures can be used here. The ripple carry adder is constructed by cascading full adders (FA) blocks in series. One full adder is responsible for the addition of two binary digits at any stage of the ripple carry. The carryout of one stage is fed directly to the carry-in of the next stage. Even though this is a simple adder and can be used to add unrestricted bit length numbers, it is however not very efficient when large bit numbers are used.

The existing work on the DLMS adaptive filter does not discuss the fixed-point implementation issues, e.g., location of radix point, choice of word length, and quantization at various stages of computation, although they directly affect the convergence performance, particularly due to the recursive behavior of the LMS algorithm. Therefore, fixed-point implementation issues are given adequate emphasis in this paper. Besides, we present here the optimization of our previously reported design to reduce 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.

To give high accuracy in area and power the ripple carry adder structures can be used here. The ripple carry adder is constructed by cascading full adders (FA) blocks in series. One full adder is responsible for the addition of two binary digits at any stage of the ripple carry. The carryout of one stage is fed directly to the carry-in of the next stage.

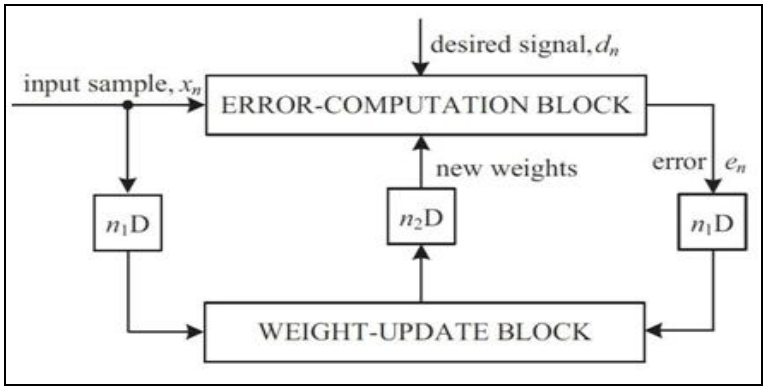


Figure 1: Structure of modified delayed LMS adaptive filter

The existing work on the DLMS adaptive filter does not discuss the fixed-point implementation issues, e.g., location of radix point, choice of word length, and quantization at various stages of computation, although they directly affect the convergence performance, particularly due to the recursive behavior of the LMS algorithm. Therefore, fixed-point implementation issues are given adequate emphasis in this paper. Besides, we present here the optimization of our previously reported design to reduce 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.

To give high accuracy in area and power the ripple carry adder structures can be used here. The ripple carry adder is constructed by cascading full adders (FA) blocks in series. One full adder is responsible for the addition of two binary digits at any stage of the ripple carry. The carryout of one stage is fed directly to the carry-in of the next stage.

2. Literature Survey

Adders form an almost obligatory component of every contemporary integrated circuit. The prerequisite of the adder is that it is primarily fast and secondarily efficient in terms of power consumption and chip area. There are various adder topologies are there such as Ripple carry adder, carry save adder, carry look ahead adder, carry increment adder, carry skip adder, carry select adder etc.

In area, power specification of the modified delayed LMS algorithm each of the adder structures has its own variations. In 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. Ripple Carry Adder

The ripple carry adder is constructed by cascading full adders (FA) blocks in series. One full adder is responsible for the addition of two binary digits at any stage of the ripple carry. The carryout of one stage is fed directly to the carry-in of the next stage. Even though this is a simple adder and can be used to add unrestricted bit length numbers, it is however not very efficient when large bit numbers are used. One of the most serious drawbacks of this adder is that the delay increases linearly with the bit length. The worst-case delay of the RCA is when a carry signal transition ripples through all stages of adder chain from the least significant bit to the most significant bit, which is approximated by:

$$t = (n-1)\,tc + ts$$

Wheretc is the delay through the carry stage of a full adder, and ts is the delay to compute the sum of the last stage. The delay of ripple carry adder is linearly proportional to n, the number of bits; therefore the performance of the RCA is limited when n grows bigger. The advantages of the RCA are lower power consumption as well as compact layout giving smaller chip area. The design schematic of RCA is shown in Figure (2). The simulation result is shown in Figure (2b).

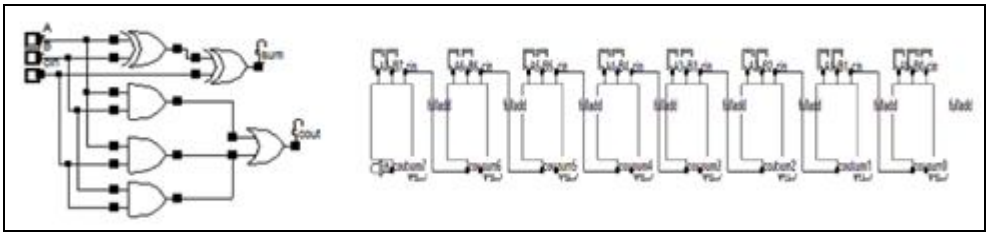


Figure 2(a): Structure of Full adder & Figure 2(b): Structure of Ripple carry adder

4. 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.

The proposed structure for error-computation unit of an *N*-tap DLMS adaptive filter consists of *N* number of 2-b partial product generators (PPG) corresponding to *N* multipliers and a cluster of *L*/2 binary adder trees, followed by a single shift–add tree.

The weight-update block performsNmultiply-accumulate operations of the form $(\mu \times e) \times xi + wi$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 only by a shift operation. Each of the MAC units therefore 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 *N* multiplications for 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 $\mu \times e$ with *L*/2, the number of 2-b digits of the input word *xi*, where the sub expression $3\mu \times e$ is shared within the multiplier. Since the scaled error ($\mu \times e$) is multiplied with the entire*N* delayed input values in the weight-update block, this sub expression can be shared across all the multipliers as well. This leads to substantial reduction of the adder complexity. The final outputs of MAC units constitute the desired updated weights to be used as inputs to the error-computation block as well as the weight-update block for the next iteration.

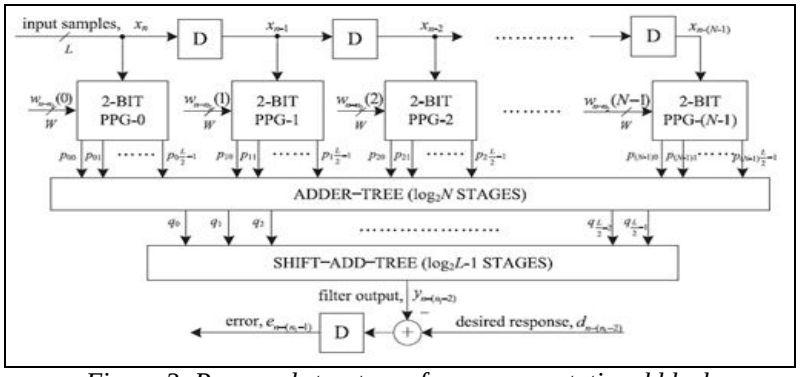


Figure 3: Proposed structure of error computational block

Implementation, we use a novel partial product generator and propose a strategy for optimized balanced pipelining across the time-consuming combinational blocks of the structure. In this paper, we find that the proposed design offers less area-delay product and less energy-delay product than the best of the existing systolic structures, on average, for filter lengths N = 8. But We propose an efficient implementation scheme of the proposed architecture for steady-state error. Moreover, we have proposed a bit-level pruning of the proposed architecture, which provides saving in ADP and saving in EDP over the proposed structure.

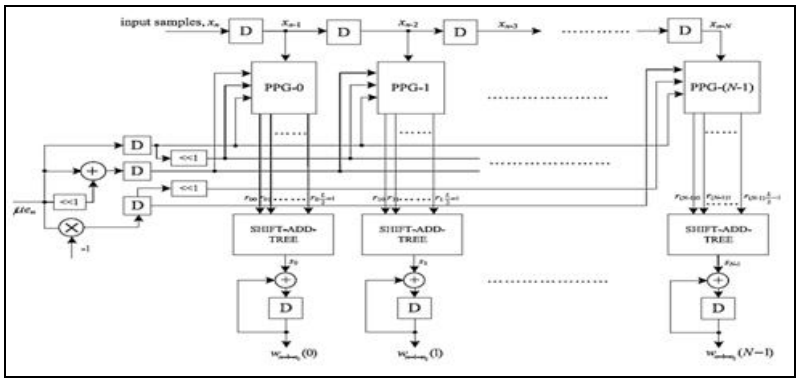


Figure 4: Proposed structure of weight update block

- **Partial Product Generation:**
  The structure of each PPG consists of 4 numbers of 2-to-3 decoders and the same number of AND/OR cells.
- **2-3 Decoders**:
  Each of the 2-to-3 decoders takes a 2-b digit as input and produces three outputs bo,b1,b2 formula for decoder:b0 = u0 . u1, b1 =. u0 · u1, and b2 = u0 · u1. The decoder output b0, b1 and b2 along with w, 2w, and 3w are fed to an AOC, where w, 2w, and 3w are in 2's complement representation and sign-extended to have (W + 2) bits each. To take care of the sign of the input samples while computing the partial product corresponding to the most significant digit (MSD), i.e., (uL−1uL−2) of the input sample, the AOC (L/2 − 1) is fed with w, −2w, and −w as input since (uL−1uL−2) can have four possible values 0, 1, −2, and −1.
- **AOCs**:
  Each AOC consists of three AND cells and two OR cells. Each AND cell takes an n-bit input D and a single bit input b, and 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. Each OR cell similarly takes a pair of n-bit input words and has n OR gates. A pair of bits in the same bit position in B and D is fed to the same OR gate. 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), respectively. The decoder along with the AOC performs a multiplication of input operand w with a 2-b digit (u1u0), such that the PPG performs L/2 parallel multiplications of input word w with a 2-b digit to produce L/2 partial products of the product word wu.
- **Adder Tree:**
  Conventionally, we should have performed the shift-add operation on the partial products of each PPG separately to obtain the product value and then added all the four product values to compute the desired inner product. However, the shift-adds operation to obtain the product value increases the word length, and consequently increases the adder size of three additions of the product values. To avoid such increase in word size of the adders, we add all the four partial products of the same place value from all the four PPGs by Ripple carry Adder tree.
- **Shift-add tree:**

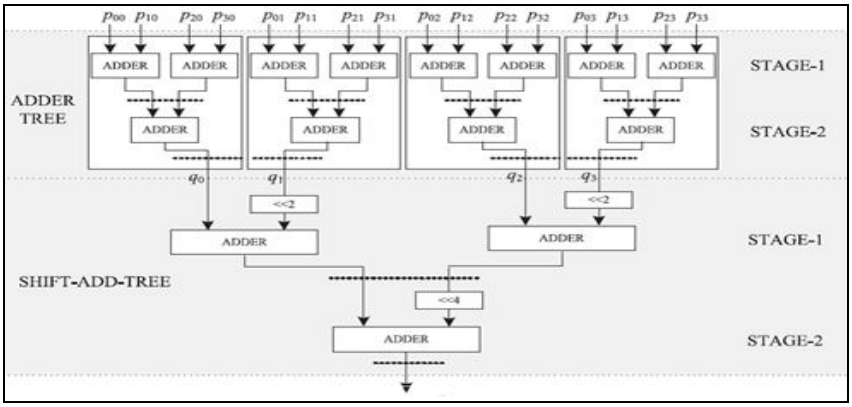


Figure 5: Adder Structures

All the FOUR partial products generated by each of the four PPGs are thus added by four binary adder trees. The outputs of the four adder trees (RCA) are then added by a shift-add tree according to their place values. Each of the binary adder trees require two stages of adders to add N partial product, and the shift–add tree to add four output of four binary adder trees.2 The addition scheme for the error-computation block for a four-tap filter and input word size L = 8 is shown in Fig. 7. For N = 4 and L = 8, the adder network requires four binary adder trees of two stages each and a two-stage shift–add tree. In this figure, we have shown all possible locations of pipeline latches by dashed lines, to reduce the critical path to one addition time. which would lead to a high adaptation delay and introduce a large overhead of area and power consumption for large values of four and eight.

5. Experimental Results and Comparison

To demonstrate the advantages of the proposed ripple carry adder we can simulate both of the carry save adder and the ripple carry adder by using Xilinx 8.1 tool.

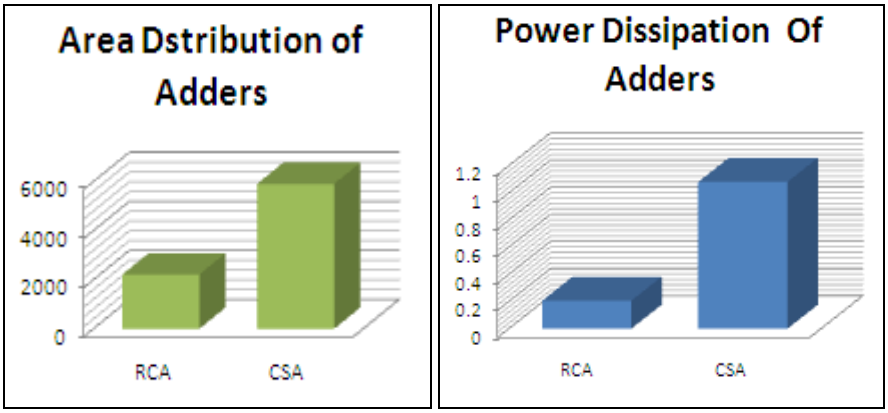
MODULE	AREA (By gate counts)	POWER (mW)
Fixed-Point LMS Adaptive Filter With Carry Save addition	5884	42
Fixed-Point LMS Adaptive Filter With Ripple Carry addition	5456	41

Table 1

5.1. Area, Delay and Power Comparison

In 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 performanceCMOS circuits. The operating frequency of adder topologies are set at 500MHz and its power dissipation and delay are observed. The graph in Figure (10a) shows the distribution of power dissipation values of different adder topology. Figure (10b, c, d) represents the area distribution, transistor count and delay distribution of adders.

From the power distribution graph it is observed that the maximum power dissipation occurs for carry select adder and next comes the carry save adder. The least power dissipation occurs for ripple carry adder and carries increment adders. From the area distribution and gate count the carry select and carry save adders occupies more area and gate count, ripple carry and carry increment occupies less area and gate count.



6. References

1. 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.
2. S. Ramanathan and V. Visvanathan, “A systolic architecture forLMS adaptive filtering with minimal adaptation delay,” in Proc.Int.Conf. Very Large Scale Integr. (VLSI) Design, Jan. 1996,pp. 286–289.
3. 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.
4. 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.
5. B vijay kumar,v. sandeep kumar “Designing ripple carry adder using a new design of the cmos full adder”international journal of computer applications e
6. ” Improved carry select adder with reduced area and low power consumption “ – padma devi ,ashima giridher,Balwindher singh
7. ’’Area,delay and power comparison of adder topologies’’ –R Uma,vidya vijayan ,m mohanapriya, Sharon Paul International journal of vlsi design & communication systems
8. Y. Sunil Gavaskar Reddy and V.V.G.S.Rajendra Prasad, “Power Comparison of CMOS and Adiabatic Full Adder Circuits”, International Journal of VLSI design & Communication Systems (VLSICS) Vol.2, No.3, September 2011

9. Pudi. V, Sridhara., K, "Low Complexity Design of Ripple Carry and Brent Kung Addersin QCA",Nanotechnology,IEEE transactions on,Vol.11,Issue.1,pp.105-119,2011
10. Suryasnata Tripathy, L B OM Prakash, B. S. Patro ,Sushanta K. Mandal " A Comparative Analysis of Different 8-Bit Adder Topologies at 45nm Technology" International Journal of Engineering Research & Technology (IJERT) Vol. 2 Issue 10, October - 2013.