

# ASIC Implementation of Low Power, Area Efficient Adaptive FIR Filter Using Pipelined DA



Grande Naga Jyothi and Sridevi Sriadibhatla

**Abstract** This paper presents a brief information on the ASIC implementation of adaptive finite impulse response (FIR) filters based on pipelined distributed arithmetic (DA) architecture. The pipelined sum of partial products of input samples is stored in the lookup table of the DA. The area of the proposed design is reduced by replacing the adder of the shift accumulation unit with the carry-save adder. The throughput rate of the design is increased by having fast clock to the carry-save adder and slow clock to the remaining circuit. The proposed design is implemented in Synopsys 90 nm CMOS technology. The area delay product (ADP), minimum cycle period (MCP), and energy per sample are reduced when compared with the conventional DA-based architectures.

**Keywords** Distributed arithmetic · LUT · FIR filter · LMS algorithm  
Adaptive filter

## 1 Introduction

Several digital signal processing systems (DSP) need linear filters to adjust the changes in the signals they process. Adaptive filters are one among them, which are significantly used in DSP applications like software-defined radio [1], channel equalization [2], noise cancelation. Adaptive filters can be designed as finite impulse response (FIR) and infinite impulse response (IIR) filters. Adaptive FIR filter has advantages than adaptive IIR filter because of their stability and easy update of

---

G. Naga Jyothi (✉) · S. Sriadibhatla  
VIT University, Vellore 21218, India  
e-mail: grande.nagajyothi@vit.ac.in

S. Sriadibhatla  
e-mail: sridevi@vit.ac.in

the filter coefficients. Transfer function of adaptive filter is controlled by variable parameters and these parameters are adjusted according to the optimized algorithms. Widrow-Hoff least mean square (LMS), recursive least square (RLS), normalized least mean square (NLMS) algorithms are some of the algorithms. The RLS adaptive algorithm involves more complicated mathematical operations and requires more computation process. NLMS algorithm is used for the improvement of the voice quality. Widrow-Hoff LMS has less complexity and easy to compute. LMS adaptive filters are more preferable for coefficient update and system stability. The LMS algorithm has good convergence property.

Adaptive filters can be implemented by using multipliers, adders, and memories. The hardware complexity of the adaptive filter will be more by using multipliers, the complexity of the design can be reduced by using multiplier-less architectures. Distributed arithmetic is one of the multiplier-less designs used for reducing the hardware complexity. DA can compute bit serial operation of vector dot product, inner partial product with high efficiency. DA uses a lookup table (LUT) for the coefficient updating and follows shift operation for output calculation. DA was first introduced by Croisier et al. [3] in the 1970s. Zohar [4] has done some document work on DA. White [5] proposed an article on vector dot product using DA and applied the concept of AGM digital autopilot. Allred et al. [6] proposed DA-based adaptive FIR filter. In the paper [6], the authors used two separate LUT architectures. One LUT is used for calculating filter output and other LUTs is used for weight updating of the filter. Guo and Brunner designed DA-based adaptive filter by using single LUT for weight update and filtering, but the design is suitable for lower order filters [7]. Meher and Park [8] proposed the pipelined DA-based adaptive FIR filter with low adaptation delay. Meher [9] proposed systolic DA-based FIR filter. Mohanty and Meher [10] proposed block LMS adaptive filter using DA.

In this paper, we proposed a novel DA-based adaptive filter with low computation delay and power consumption. The power consumption is substantially decreased by reducing the delays in the pipeline architecture of DA. The throughput rate of the design is increased due to use of carry-save accumulator instead of using shift accumulator. By using the proposed architecture, coefficient updating and filtering can be performed in parallel.

The remnant of the paper is as follows. In the next section, we briefly explain the overview of the LMS adaptive algorithm and describe the mathematical calculation of DA and background of LMS adaptive filter using DA. Section 3 describes the proposed novel DA-based adaptive filter. Section 4 provides the synthesis results of the proposed and existing architectures. Finally, conclusions are given in Sect. 6.

## 2 Overview of LMS Adaptive Filter

The basic LMS adaptive FIR filter architecture is as shown in Fig. 1. For every cycle, the adaptive filter evaluates error value and the filter output. The obtained error value is used for updating the filter coefficient.

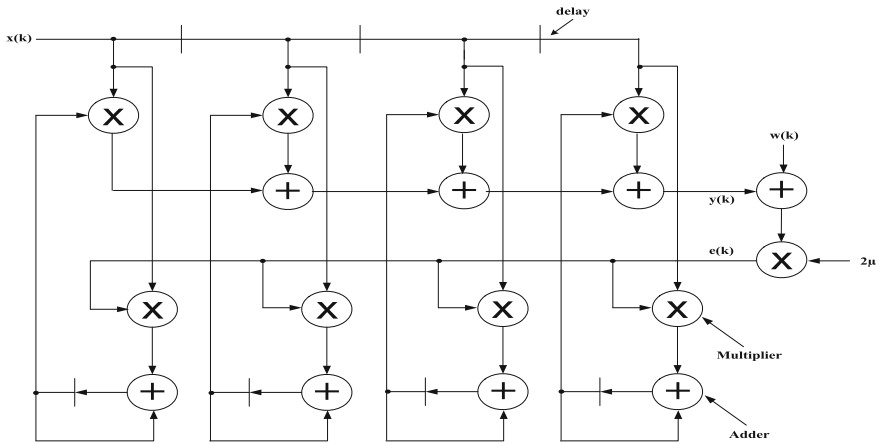


Fig. 1 Basic adaptive filter

Input signal vector  $X(k)$  is represented as

$$X(k) = [x(k), x(k - 1), \dots, x(k - K + 1)]^T \tag{1}$$

where  $X(k)$  is the input signal at time 'k' and 'T' is the transpose of the vector. The output signal  $Y(k)$  of an adaptive filter can be represented as

$$Y(k) = X^T(k)d(k) \tag{2}$$

where  $d(k)$  is the filter coefficient vector and it is represented as

$$d(k) = [d_0(k), d_1(k), d_2(k), \dots, d(K - 1)(k)]^T \tag{3}$$

The Window's LMS algorithm for weight updating can be represented as

$$d(k + 1) = d(k) + \mu e(k)X(k) \tag{4}$$

where  $e(k)$  is the error signal,  $\mu$  is the step size parameter which determines the convergence speed and accuracy of the filter. The error signal can be obtained by the difference between the desired signal and the output signal. The equation is given as

$$e(k) = D(k) - Y(k) \tag{5}$$

where  $D(k)$  is the desired signal. The input signal of the filter  $X(k)$  is nourished into the delay line and then shifted to the right for each sampling period.

## 2.1 Background of DA

DA is an efficient architecture for computing inner product vectors with an efficient mechanism. Let us consider the inner product as

$$y = \sum_{k=0}^K d_k x_k \quad (6)$$

where  $d_k$  is fixed coefficient,  $x_k$  is the input signal and  $K$  is the number of input words.  $x_k$  is a 2's complement binary number scaled such that  $|x_k| < 1$ , then  $x_k$  and  $-x_k$  can be expressed as

$$x_k = -b_{k0} + \sum_{l=1}^{L-1} b_{kl} 2^{-l} \quad (7)$$

where  $x_k = b_{k0}, b_{k1}, \dots, b_{k(L-1)}$ .

On substituting  $x_k$  in Eq. (6) and interchanging the summation order; finally, we get the equation as follows:

$$y = \sum_{k=1}^K d_k \left[ -b_{k0} + \sum_{l=1}^{L-1} b_{kl} 2^{-l} \right] \quad (8)$$

Equation (8) is the normal inner multiplication expression. By changing the summation order, finally we get equation as follows.

$$y = \sum_{l=1}^{L-1} \left[ \sum_{k=1}^K d_k b_{kl} \right] 2^{-l} + \sum_{k=1}^K d_k (-b_{k0}) \quad (9)$$

Equation (9) gives computation of distributed arithmetic and the inner product is computed by

$$y = \sum_{n=1}^{N-1} [2^{-n} C_n - C_0] \quad (10)$$

where  $C_n = \sum_{k=0}^{K-1} [d(k)b(k)_n]$ .

## 3 DA-Based Adaptive Filter Architecture

The DA-based adaptive FIR filter architecture [8] of filter length  $K = 4$  is as shown in Fig. 2. It consists of a four-point inner product block, weight updating block, control unit block. Along with it, error calculation  $e(n)$  and sign magnitude controller units are present.

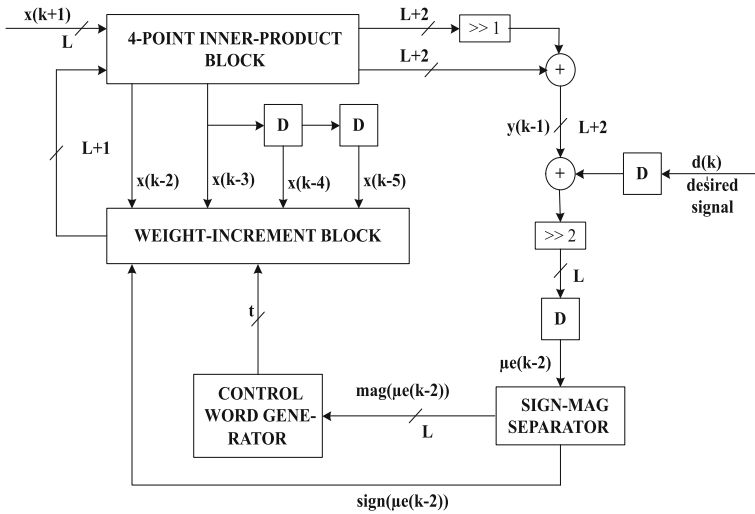


Fig. 2 DA-based LMS adaptive filter

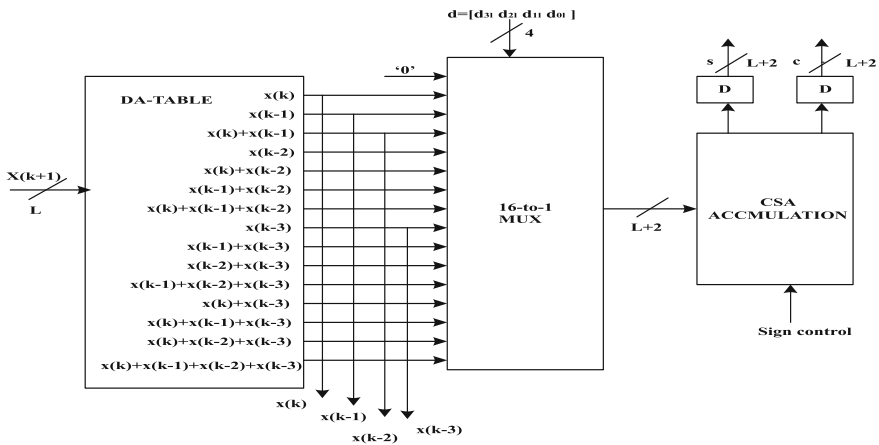


Fig. 3 Four-Point inner product block

The four-point inner product block consists of DA table, multiplexers, XOR gates, and carry-save accumulator as shown in Fig. 3. The DA table has 16 registers, which store 16 combinations of inner product values as shown in Fig. 5 and fed as input to the 16:1 multiplexer. The feedback filter coefficients from the weight updating block will act as selection lines to the multiplexer and the output from the multiplexer is given to the carry-save accumulator unit. The output from MUX is given to the XOR gate for sign control. If the MSB bit of MUX is zero, then normal addition operation can be performed else 2's complement addition can be performed. After 'K' clock cycles, the carry-save accumulator gives sum and carry outputs. By using fast bit

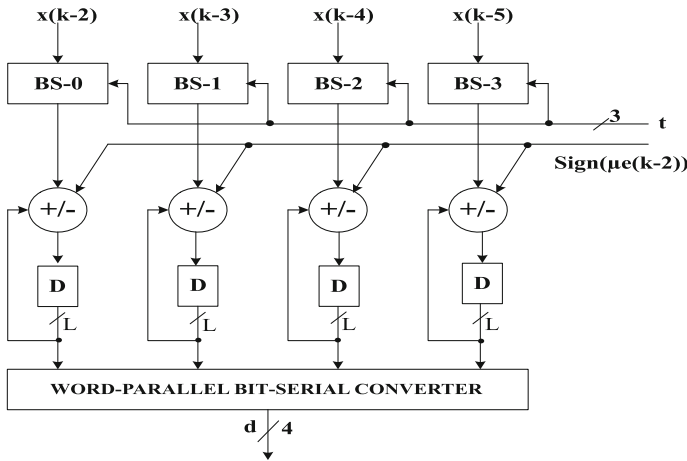


Fig. 4 Weight increment block

clock cycle to carry-save adder accumulator, the throughput rate of the architecture is increased. The generated sum and carry are added to obtain final filter output  $y(k)$ . The filter output  $y(k)$  is subsequently subtracted from the desired signal  $D(k)$  to achieve error sample  $e(k)$ .

The weight increment block consists of four barrel shifters, four adder/subtract units and a word parallel bit serial converter as shown in Fig. 4. The multiplication of the input  $x_k$  with error  $e(k)$  can be performed by using barrel shifter and the output from the barrel shifter is added/subtracted with current weights to get weights updation. The updated weights are given as selection lines to the 16:1 multiplexer of the four-point inner product unit.

### 4 Proposed Pipelined DA-based Adaptive FIR filter

In the basic DA-based adaptive FIR filter, the inner products are computed using DA table with registers and adders. The inner product of DA table is shown in Fig. 5. The input  $x(K + 1)$  with length  $L$  is fed to the register and achieves the output  $x(k)$ . Again,  $x(k)$  is given to another register to get  $x(k - 1)$ . The input samples  $x(k + 1)$  and  $x(k)$  are added and passed through the register to produce  $x(k) + x(k - 1)$ . Similarly, the registers and adders will generate the DA table outputs  $x(k - 2)$ ,  $x(k) + x(k - 2)$ ,  $x(k - 1) + x(k - 2)$ ,... and so on and these 16 outputs are passed as inputs to the 16:1 multiplexer with filter coefficient as selection lines. The DA table outputs are achieved by using 15 registers and 7 adders. These registers occupy more area and consume more power. To reduce the area, we proposed a novel pipelined DA table as shown in Fig. 6.

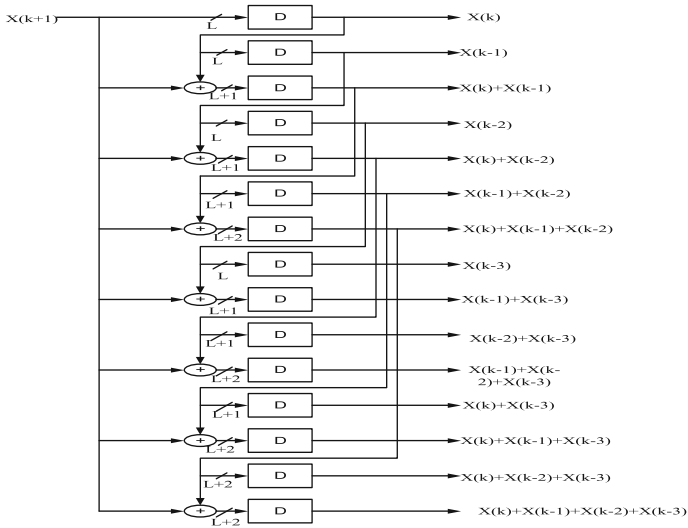


Fig. 5 DA table for generation of sum of input samples

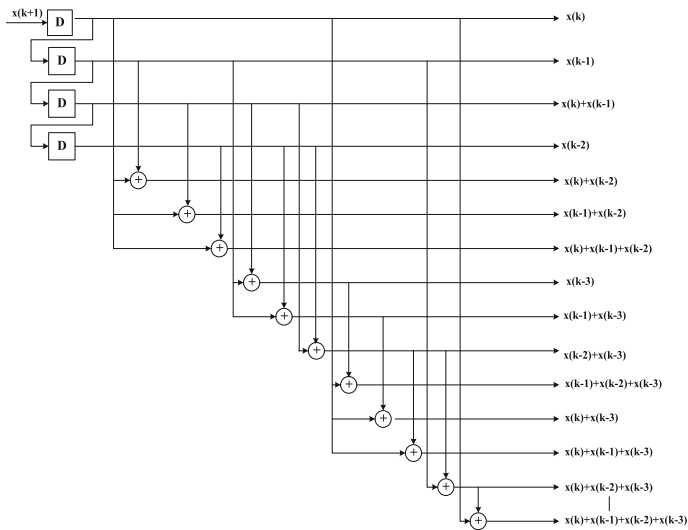


Fig. 6 DA table for proposed pipelined DA FIR Filter

In the proposed pipelined DA table design, instead of passing  $x(k + 1)$  input samples every time to the adders and registers we make use of previous register samples obtained from the registers and fed back them as inputs to the adder to achieve the same DA table characteristic. By using this concept, inner product of the DA table is generated with 4 delays and 10 adders. This causes the reduction

of 11 registers with increase of 3 adders. By using the proposed design, the area of DA-based adaptive filter can be reduced.

### 5 Simulation Results

The proposed work is coded in Verilog HDL and implemented in ASIC 90 nm CMOS library in Synopsys design compiler. The performance comparison of proposed pipelined DA-based adaptive FIR filter with systolic DA architecture of Ref. [9] and DA-based adaptive FIR filter of Ref. [8] are presented in Table 1. The proposed architecture produces more number of outputs per cycle when compared with the existing architectures. The area occupied by the proposed pipelined DA architecture is less when compared with the systolic DA architecture [9] and the proposed architecture occupies less area when compared with the architecture of [8]. The area delay product (ADP), power delay product and minimum cycle period (MCP), maximum sampling period (MSP) are low when compared with the two DA architectures. The simulation results for 4-tap pipelined DA-based adaptive FIR filter are shown in Fig. 7. The layout diagram for the proposed pipelined DA-based adaptive FIR filter is shown in Fig. 8.

**Table 1** Performance comparison of ASIC synthesis result for CMOS 90 nm technology for an 16-tap FIR Filter

Design	Area	ADP	PDP	MSP	MSF	Power
Filter [8]	18,257	94,936	32.38	5.20	192	6.2269
Filter [9]	71,195	127,439	60.76	1.79	558	33.944
Proposed filter	17,163	32,757	30.03	5	200	6.006

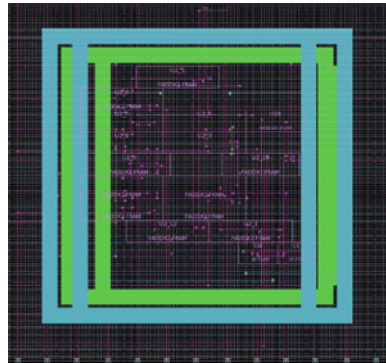
Units Area Sq.  $\mu\text{m}$ , ADP Sq.  $\mu\text{m} \times \text{ns}$ , PDP  $\text{mW} \times \text{ns}$ , MSP ns, MSF MHz, Power mw



**Fig. 7** Simulation results for 4-tap pipelined DA-based adaptive FIR filter



**Fig. 8** Layout chip diagram for the proposed pipelined DA-based adaptive FIR filter with 90 nm technology



## 6 Conclusion

We have designed and implemented pipelined DA-based adaptive FIR filter for high throughput, low power, and low area. The throughput for the proposed architecture is increased by using fast clock pulse to the carry-save adder of the accumulator. The reduction in the number of delays of the pipelined architecture of DA table has reduced the area. Weight updation and LUT updation are performed in parallel. The proposed architecture has 14% less in power and 30% less in area when compared with the basic DA-based adaptive filters.

## References

1. Hentschel MH, Fettweis G (1999) The digital front—end of software radio terminals. *IEEE Personal Commun Mag* 6(4):40–46
2. Vaidyanathan PP (1993) *Multirate systems and filter banks*. Prentice Hall, Englewood Cliffs, NJ
3. Croisier A, Esteban DJ, Levilion ME, Rizo V (1973) Digital Filter for PCM Encoded Signals. U.S. Patent 3,777,130, 4 Dec 1973
4. Zohar S (1973) New hardware realization of non recursive digital filters. *IEEE Trans Comput* C-22:328–338
5. White SA (1989) Applications of distributed arithmetic to digital signal processing : a tutorial review. *IEEE ASSP Mag* 6(3):419
6. Alled DJ, Yoo H, Krishnan V (2005) LMS adaptive filters using distributed arithmetic for high throughput. *IEEE Trans Circuit syst* 52(7):1327–1337
7. Guo R, Debrunner LS (2011) Two high-performance adaptive filter implementation schemes using distributed arithmetic. *IEEE Trans Circuits syst II Exp. briefs* 58(9):600–604
8. Meher PK, Park SY (2011) High-throughput pipelined realization of adaptive FIR filter based on distributed arithmetic. In: *Proceedings of 2011 IEEE/IFIP 19th International Conference on VLSI, System-on-Chip, (VLSI-SOC11)*. Oct 2011, pp 428–433

9. Meher PK (2006) Hardware-efficient systolization of DA-based calculation of finite digital convolution. *IEEE Trans Circuits Syst II* 53(8):707–711
10. Mohanty BK, Meher PK (2013) A high-performance energy-efficient architecture for FIR adaptive filter based on new distributed arithmetic formulation of block LMS algorithm. *IEEE Trans Signal Process* 61(4):921–932