POWER EFFICIENT AND HIGH THROUGHPUT OF FIR FILTER USING BLOCK LEAST MEAN SQUARE ALGORITHM IN FPGA

M.Devipriya[1], V.Saravanan[2], N.Santhiyakumari[3]

[1]PG scholar, Department of Electronics and communication Engineering, Knowledge institute of technology, Anna University, Salem, Tamilnadu, India.
[2]Associate professor, Department of Electronics and communication Engineering, Knowledge institute of technology, Anna University, Salem, Tamilnadu, India
[3]Professor & Head, Department of Electronics and communication Engineering, Knowledge institute of technology, Anna University, Salem, Tamilnadu, India.

Abstract

In silicon on chip technology demands high performance and low power Very Large Scale Integrated Circuit (VLSI) digital signal processing (DSP) systems. The aim of this paper explores the power consumption technique for the architecture of Finite Impulse Response (FIR) adaptive filter. An adaptive FIR filter with Block Least Mean Square (BLMS) algorithm was developed to reduce the power. Distributed arithmetic (DA)-based formulation of BLMS algorithm is used to reduce the area where both convolution operation to compute filter output and correlation operation to compute weight-increment term could be performed by using the same LUT. Thus a DA based implementation of adaptive filter is highly computational and area efficient.

Keywords: *FIR, FPGA, DSP, DA, VLSI, BLMS.*

----------***----------

I. INTRODUCTION

Finite impulse response (FIR) filters are the most popular type of filters implemented in software. FIR Filter plays an important role in digital signal processing which could be used to as the Low Pass Filter, Pass Band Selection, Anti-Aliasing, Extraction and Interpolation and etc. In digital signal processing, traditional filter is realized by high speed multiplication accumulator where only limited operation can be done during the next sampling period, so the bandwidth is limited and the speed is low. FIR filter is usually implemented by using a series of delays, multiplexer, and adders to create the filter's output. An adaptive FIR filter with Block Least Mean Square (BLMS) algorithm was developed to reduce the power. Distributed arithmetic (DA) based formulation of BLMS algorithm is used to reduce the area where both convolution operation to compute filter output and correlation operation to compute weight-increment term could be performed by using the same LUT. Thus a DA based implementation of adaptive filter is highly efficient**.**

1.1 Prior Work

FIR filters are normally used in abundance in many consumer level products especially those related with image processing and voice/video communication [1]. To provide appropriate response, FIR filters may follow more than one stage. To implement High Speed Digital Signal Processor systems, on field programmable gate array (FPGA) are usually used as hardware platform. There is another implementation for this purpose, which is Application Specific Integrated Circuits (ASIC) used overwhelmingly. FPGA has the major advantages over traditional DSPs and ASICs in terms of project cost, flexibility, reconfigures ability and reliability [2]. More recently, structured ASIC technology has given lower cost solutions to full custom ASIC by predefining several layers of silicon functionality that require the definition of only a few fabrication layers to implement the required design. However, the FPGA platform provides high performance and flexibility with the option to reconfigure. In wireless communication system, realization of FIR filters as a part of DSP depends upon the use [3]. A decrease in performance is noticeable when FIR filters use some internal components implemented on FPGA. So we have used the distributed arithmetic algorithm instead of multiply and Accumulative Unit (MAC) to increase performance of FIR Filter. An adaptive filter is a filter that self-adjusts its transfer function according to an optimizing algorithm. Because of the complexity of the optimizing algorithms, most adaptive filters are digital filters that perform digital signal processing and adapt their performance based on the input signal. Adaptive filter are required when either the fixed specifications are unknown or time invariant filters cannot satisfy the specifications.

Adaptive filters are time-varying since their parameters are continually changing in order to meet a performance requirement. The hardware implementation requires several of performances such as high speed, low power dissipation, small

chip area and good convergence characteristics. However it is difficult to satisfy these characteristics simultaneously, so efficient algorithms and efficient architectures are desired. The design of adaptive filter algorithm is an important part within the design of adaptive filter. Among many adaptive filter algorithms, the least-mean-square (LMS) algorithms have been used because of their relatively small computational complexity of 2L, where L is the filter length. The shortcoming of the LMS algorithm is that it has slow convergence rate for input signal such as speech. Block processing is an effective approach to reduce the computational complexity and is employed in the LMS. The block least mean square (BLMS) algorithm is one of the fastest and computationally efficient adaptive algorithms. The popularity of the FPGA is due to balance that FPGAs provide the designer in terms of flexibility, cost, and time-to-market. In this paper, the design of a BLMS algorithm based adaptive filter will be analyzed.

2. FPGA BASED SIGNAL PROCESSING

Most digital signal processing done today uses a specialized microprocessor, called a digital signal processor, capable of very high speed multiplication. This traditional method of signal processing is bandwidth limited. There occur a fixed number of operations that the processor can perform on a sample before the next sample arrives. FPGA-based digital signal processing is based on hardware logic and does not suffer from any of the software based processor performance problems [4]. FPGAs allow applications to run in parallel so that a 128 Tap filter can run as fast as a 10 tap filter. Applications can also be pipelined in an FPGA, so that filtering, correlation, and many other applications can all run simultaneously. In an FPGA, most of the application is working mostly when timing requirements are strict. An FPGA can offer 10 to 1000 times the performance of the most advanced digital signal processor at similar or even lower costs. Nowadays, the use of FPGAs is increasing. They are the prototyping hardware devices, combining the main advantages of ASIC and DSP processors, since they provide both a programmable and a dedicated hardware solution. It is usual to use an FPGA as the prototyping device due to factors such as time and cost. Moreover, an FPGA is more efficient in power consumption, an advantage for battery-operated systems, and, for the same application, requires less clock system speed compared to a DSP or a general-purpose processor, offering better electromagnetic compatibility properties. Thus, for a wide range of applications, FPGA implementation might be the best option. However, in the case of low sampling frequency requirements or no low power consumption needs, some other devices could be more suitable. In this paper, FPGAs are the target hardware used. FPGAs are ideally suited for the implementation of adaptive filters. However, there are several issues to be addressed. Calculation of adaptive algorithm in software simulations are normally carried out with floating point precision. Another concern is the filter tap itself. Various techniques have been devised to efficiently calculate the convolution operation when the filters coefficients are fixed in advance. For an adaptive filter whose coefficients change over time, these methods will not work or need to be modified significantly.

2.1. Hardware Complexity

Due to the high performance requirements and increasing the complexity of DSP and multimedia communication applications, filters with large number of taps are required to increase the performance in terms of high sampling rate. As a result the filtering operations are computationally intensive and more complex in terms of hardware requirements. The FIR filters perform the weighted summations of input sequences with constant coefficients in most of the signal processing and multimedia applications. These filters are widely used in video convolutions functions, signal preconditioning, and other communication applications. The decrease in computational complexity causes the increase in the performance, in terms of speed, area and power. High speed, low area and power efficient conscious design techniques in Silicon on chip include efforts at all level of abstraction. One way to efficiently incorporate high performance design technique is to implement IP cores. These cores have following major advantages.

- Reusability across designs
- Reduction of the design effort
- Shorter time to market.

The disadvantage of FIR filters is that they require high order. The high order demands more hardware, area and power consumption. To minimize these parameters, our goal is to implement an efficient high order filter in digital systems. By the reduction of arithmetic in terms of multipliers, our goal is to reduce the parameters namely, hardware, area and power. This is the ultimate goal of the implementation of an efficient FIR filter and hence DA algorithm is used for implementation of high order FIR filter. FIR filter is incorporated with a MAC unit. The purpose of MAC unit is to multiply the input with constant coefficients, to shift and then to add them. This process is repeated until all partial products produce the output after accumulation. It increases the hardware complexity because a simple multiplier circuitry is used. The idea is to somehow bypass or replace the multiply and shift operations with less complex operations. Distributed Arithmetic (DA) can be used to replace MAC unit. The DA Algorithm actually uses lookup table for storing constant coefficients. So the use of lookup tables reduces the hardware complexity and hence the new design is more efficient in terms of less area, more speed and low power consumption. FIR filter reference core uses a simple MAC unit. We have replaced MAC unit in FIR filter reference core with DA Algorithm. In this study, performance of Reference Core with Simple MAC and reference core with DA is compared.

3. DA TECHNIQUE

Distributed arithmetic is a bit level rearrangement of a multiply accumulate to hide the multiplications. It is a powerful technique for reducing the size of a parallel hardware multiply-accumulate that is well suited to FPGA designs. It can also be extended to other sum functions such as complex multiplies, Fourier transforms and so on. In most of the multiply accumulate applications in signal processing, one of the multiplicands for each product is a constant. The DA targets the products of sums which cover all filtering application and frequency transfer functions. DA uses Look-Up Table (LUT) which stores the constant coefficients of FIR Filter. The size of Look-Up Table (LUT) in DA algorithm is 2 , where k is the number of filter taps. When number of taps increases, LUT grows exponentially. By using offset Binary Code (OBC), the size of the LUT can be reduced. This is very efficient in terms of less hardware and more speed. Many DSP applications required FIR which having MAC (Unit multiplier and add accumulator), replacing MAC with LUT-Based DA algorithm having power, efficiency and less area usage. Proposed DA algorithm is hardware efficient for VLSI and FPGA, but LUT-Less OBC is efficient only for custom VLSI [5].We have used DA for multiplier less architecture in FPGA. For DA based on look-up table having constant coefficient and changing variable, one needs to design a highly efficient FIR in digital signal processing. DA can be used for high order filter. There are two techniques used in DA algorithm, one of which is parallel distributed and the other is serial distributed. [6].The DSP FIR filter functions are used in communication(e.g. Telecommunication in Biomedical Signal Processing, Communication, Wireless satellite and Image processing) which is performed efficiently. The multipliers in MAC unit of many DSP functions have more power and area requirements. There are two techniques in this respect which are multiplier less. One of them is Conversion based, in which coefficients of filters are converted into numeric representation. The second is based on LUT which stores pre-computed coefficients values of FIR filters. The LUT in DA algorithm uses more memory. [7].

4. METHODOLOGY

By using adaptive filter applications, such as adaptive echo cancellation and adaptive noise cancellation, require adaptive filters with a large filter length. The standard LMS algorithm to the adaptive filter, this algorithm might take a long time to complete the filtering and coefficients updating process. This length of time might cause problems in these applications because the adaptive filter must work in real time to filter the input signals. In this situation, we can use LMS algorithm.
The block LMS algorithm differs from the standard LMS algorithm in the following ways:

- The block LMS algorithm updates the coefficients of an adaptive filter block by block. The block size is exactly the same as the filter length. However, the standard LMS algorithm updates the filter coefficients sample by sample.
- The block LMS algorithm requires fewer multiplications than the standard LMS algorithm. If both the filter length and block size are N, the standard LMS algorithm requires N ($2N$+1) multiplications, whereas the fast block LMS algorithm requires only ($10N\log_2 N$+$26N$) multiplications. If N = 16, block LMS algorithm can execute 16 times faster than the standard LMS algorithm.

The block LMS algorithm calculates the output signal and the error signal before updating the filter coefficients. This filtering method is efficient in occupy memory and also reduce power in terms of the lookup table size also minimum.

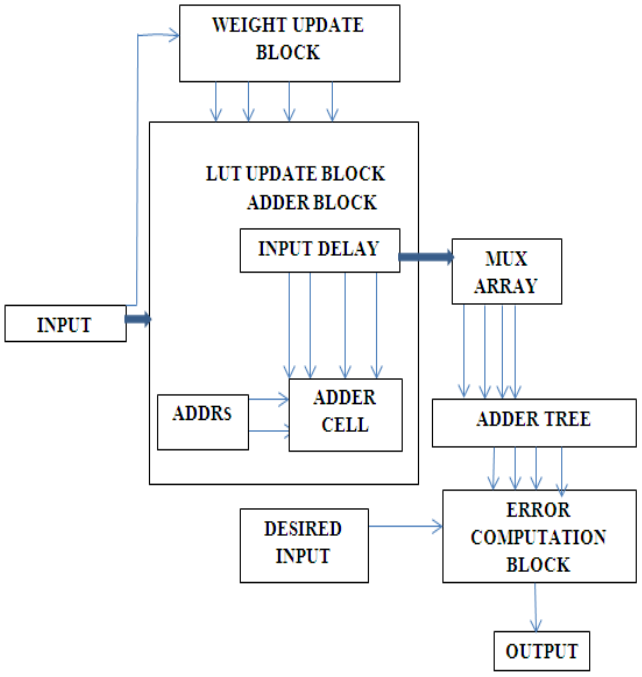

Fig 1 block diagram for DA-BLMS architecture

Figure 1.shows the block diagram of DA –BLMS is used to reduce the inner product terms.

5. IMPLEMENTATION OF DA

FIR filter has 16-taps. Each tap has 16-bit input data width and 16 filter coefficients. In designing FIR filter using DA (Distributed Arithmetic), these coefficients are placed in a look-up table. This is because these coefficients are constants. The look-up table grows exponentially when the filter coefficients are increased, so the break-up in the design is necessary and one must place four-coefficients in each look-up table. The width of each coefficient may be 8-bits or 16-bits depending upon the design. The width of the inputs data also vary to 8-bits and 16-bits, each LSB bits of input data

combined in parallel to form the address of the look-up table. Distributed arithmetic Algorithm replaces "AND" and "add" operation as compared with MAC unit. The four-look-up table store 16 coefficients of FIR filter. More than four look-up tables are used for storing more coefficients for the better response of the FIR filter. The LUTs in DA algorithm uses the multiplier less technique. The LUTs used less CLB (configuration logic blocks) in the FPGA to increase the throughput and data rates. The FPGA has no multiplier and can be used SRAM based DA algorithm. Single FPGA chip can be used instead of using multiple DSP devices for better performance in terms of speed area and power, due to SRAM present in FPGA, FPGA is more efficient for the implementation of signal processing applications. DA became best algorithm relating to filtering operation, because SRAM based FPGA stored look-up table values which are pre-computed and also FPGA gives surrounding logic in a single chip. Distributed arithmetic algorithm gives best performance when we used in filtering operation because hardware complexity less in DA as compared to conventional MAC. This operation can done in a FPGA in order to reduce the error in the FIR filter.

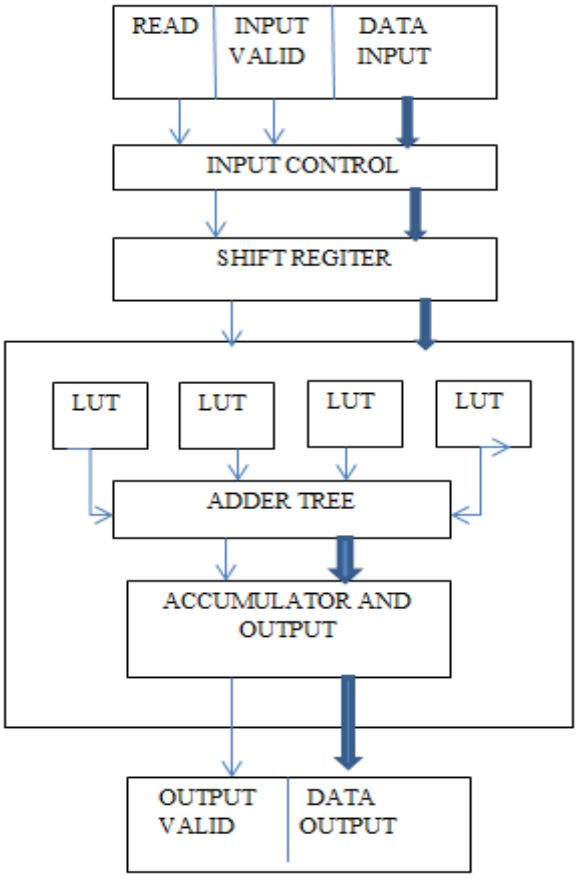

Fig 2 Implementation of Distributed arithmetic

6. SIMULATION RESULTS

The results show that distributed arithmetic algorithm is better for FIR filters implementation on FPGAs. The efficiency in terms of area, speed and power has been analyzed. The results clearly show that efficiency in terms of power dissipation and speed has been increased having almost same area consumption. The DA has two techniques, one of which is the serial DA and other one is the parallel DA. In this paper, the serial distributed arithmetic is used to make the FIR Filter more efficient. In future, the parallel DA can be used to increase the efficiency of FIR Filter in terms of data rates. The implementation of DA based algorithm, serial distributed arithmetic algorithm and parallel distributed arithmetic use the look up table.

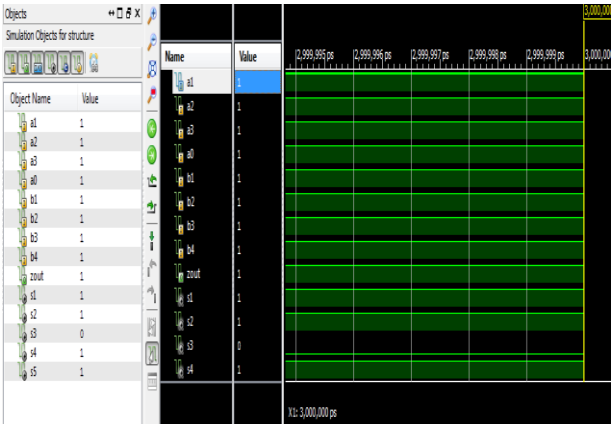

Fig 3 Simulation result for Adder cell for LUT update.

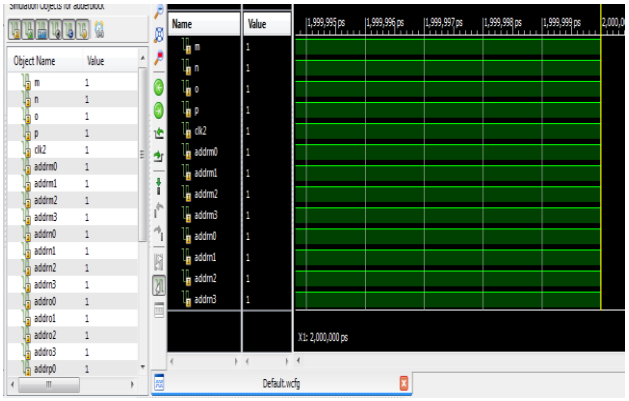

Fig 4 Simulation result for Adder block for LUT update.

Figure 3 and 4 are Simulation result for adder block .it consist of adder cell and delay unit and used to reduce the delay area.

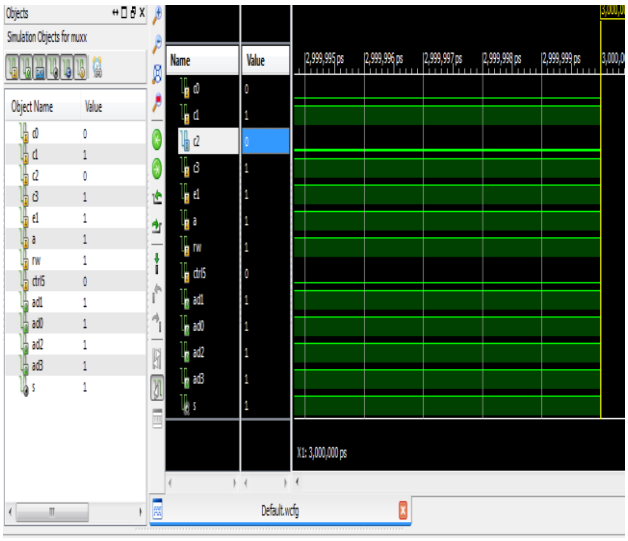

Fig 5 Simulation result for Mutiplexer array

Figure 5 simulation result for MUX 2 to 1 consist of many input and 1 output.

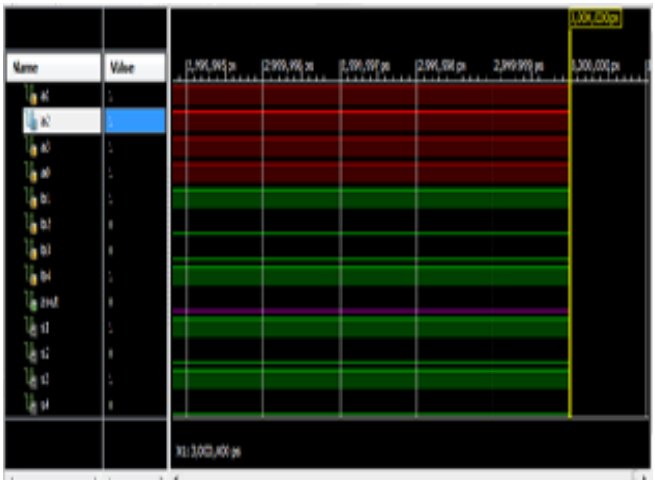

Fig 6 Simulation result for weight update block

Figure 6.simulation result for weightupdate block is used to reduce the minimum error.The size of the look up table increases when the number of filter taps is increased. For better performance of FIR Filter, LUT less DA implementation uses MUX. Every shift register uses MUX that select 0 or filter coefficients, and this technique can be used to increase the efficiency of the FIR Filter in future.

7. CONCLUSIONS

In this paper a new programmable architecture of distributed arithmetic based FIR filter is explained in detail. By using block least mean square method, we can reduce the inner products which will reduce the block size and hence the area is reduced smaller. Also number of look up table words to be updated per output is minimized. This helps to save external logic and power consumption. Compared to the available designs, this new approach reduces area delay product (ADP) up to 35% for filter length 64. The simulation result shows that the proposed method gives better performance than the existing methods in terms of both area and power consumption. This improved performance is further accurately achieved at the cost of a slightly increased computational complexity.

REFERENCES

[1] D.J.Allred,H.Yoo,V. Krishnan, W. Huang, and D.V. Anderson, "LMS adaptive filters using distributed arithmetic for high throughput,"*IEEE Trans. Circuits Syst.*, vol. 52, no. 7, pp. 1327–1337 2005.

[2] D.J.Allred, H. Yoo, V. Krishnan, W. Huang, and D. V. Anderson, "An FPGA implementation for a high throughput adaptive filter using distributed arithmetic," in *Proc. 12th Annu. IEEE Symp. Field-Programmable Custom Comput.* pp. 324–325 2004.

[3] Basant K.Mohanty and Pramod Kumar Meher,"A High-Performance Energy- Efficient Architecture for FIR Adaptive Filter Based on New Distributed Arithmetic Formulation of Block LMS Algorithm" *IEEE transactions on signal processing, vol.61, no.4 .2013*

[4] S. Baghel and R. Shaik,"FPGA implementation of fast block LMS adaptive filter using distributed arithmetic for high-throughput," in *Proc. Int. Conf. Commun. Signal Process. (ICCSP) 2011*.

[5] S. Baghel and R. Shaik, "Low power and less complex implementation of fast block LMS adaptive filter using distributed arithmetic," in *Proc IEEE Students Technol. Symp 2011.*

[6] D.P. Das, G. Panda, and S.M. Kuo,"New block filtered X LMS algorithms for active noise control systems," *IEE Signal Procesd.*, vol.1,no. 2, pp. 73–81.

[7] S. C. Douglas, Q. Zhu, and K. F. Smith, "A pipelined LMS adaptive FIR filter architecture without adaptive delay," *IEEE Trans. Signal Process 1998*.

[8] R. Guo and L. S. DeBrunner,(2011) "Two high performance adaptive filter implementation schemes using distributed arithmetic," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 58, no. 9, pp. 600–604.

[9] R.Jayashri,H.Chitra,H. Kusuma, (2011)A. V. Pavitra, and V. Chandrakanth,"Memory based architecture to implement simplified block LMS algorithm on FPGA," in *Proc. Int. Conf. Commun. Signal Process. (ICCSP).*

[10] W. Huang and D. V. Anderson,"Adaptive filters using modified sliding-block distributed arithmetic with offset binary coding," in *Proc. IEEE In. Conf. Acoust., Speech, Signal Process. (ICASSP)2009*.

[11] S. Haykin and B. Widrow,*Least-Mean-Square Adaptive Filters*. Hoboken, NJ: Wiley- Interscience 2003.

[12] B. K. Mohanty and P. K.Meher, "Delayed block LMS algorithm and concurrent architecture for high-speed implementation of adaptive FIR filters," presented at the IEEE Region 10 TENCON2008 Conf, India 2008.

[13] Q. Shen and A. S. Spanias, "Time and frequency domain block LMS algorithm for single channel active noise control," *Control Eng 1996*

[14] L. D. Van and W. S. Feng, "An efficient architecture for the DLMS adaptive filters and its applications," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.* vol. 48, no. 4, pp. 359–366.2006.

[15] L. D. Van and. S. Feng, "Efficient systolic Architectures for 1-D and 2-D DLMS adaptive digital filters," in *Proc. IEEE Asia Pacific Conf. Circuit 2000*

[16] V. Visvnathan and S. Ramanathan,"A modular systolic architecture for delayed least mean square adaptive filtering," in *Proc. IEEE Int.Conf. VLSI Des.*, pp. 332–337 ,1995