INTERNATIONAL JOURNAL OF RESEARCH
In Advanced Engineering Technologies

Volume 5, Issue 3 JUNE 2016

Design and Implementation of Distributed Arithmetic-Based Reconfigurable FIR Digital Filter

[(1)]V. Sowmya, PG Scholar in VLSI, VITS, Proddatur.
[(2)] Mr. C.Md.Aslam, M.Tech (PhD), Associate Professor, VITS, Proddatur.
[(1)]vundelasowmya9@gmail.com,[(2)]aslamhodece@gmail.com .

ABSTRACT:

A distributed arithmetic (DA)-basedmethodologies for high-throughput reconfigurable usage offinite impulse response (FIR) channels whose channel coefficients change during runtime.Expectedly, forreconfigurable DA-based usage of FIR filters, the lookuptables (LUTs) are obliged to be executed in RAM and theRAM-based LUT is discovered to be immoderate for ASICexecution. Along these lines, a mutual LUT outline isproposed to understand the DA reckoning. As opposed toutilizing separate registers to store the conceivableconsequences of fractional internal items for DA preparingof diverse bit positions, registers are shared by the DA unitsfor bit cuts of distinctive weightage. The proposed designhas almost and less delay and less area than the DA-basedsystolic structure and the carry slave adder (CSA)-basedstructure, separately, for the ASIC usage. A disseminatedRAM-based outline is additionally proposed for the fieldprogrammablegate Array (FPGA) execution of thereconfigurable FIR filter, which backings up to 91 MHzinfo testing recurrence and offers less quantity of cuts than the systolic structure and the CSAbasedstructure, individually, when executed in the XilinxVirtex-5 FPGA gadget (XC5VSX95T-1FF1136)

Keywords: Finite impulse response (FIR) filter, systolic array, field programmable gate arrays (FPGA), application specific integrated chip (ASIC), distributed arithmetic, reconfigurable implementation

I. INTRODUCTION

Finite impulse response (FIR) digital filters are extensively used due to their key role in various digital signal processing (DSP) applications [1], [2]. Along with the advancement in very large scale integration (VLSI) technology as the DSP has become increasingly popular over the years. Since the complexity of implementation grows with the filter order and the precision of computation, real-time realization of these filters with desired level of accuracy is a challenging task. Several attempts have, therefore, been made to develop dedicated and reconfigurable architectures for realization of FIR filters in application specific integrated circuits (ASIC) and field programmable gate arrays (FPGA) platforms. Systolic designs represent an attractive architectural paradigm for efficient hardware implementation of computation-intensive DSP applications, being supported by the features like simplicity, regularity and modularity of structure. Additionally, they also possess significant potential to yield high-throughput rate by exploiting high-level of concurrency using pipelining or parallel processing or both [3]. To utilize the advantages of systolic processing, several algorithms and architectures have been suggested for systolization of FIR filters [4]–[7]. However, the multipliers in these structures require a large portion of the chip-area, and consequently enforce limitation on the maximum possible number of processing elements (PEs) that can be accommodated and the highest order of the filter that can be realized. Multiplierless distributed arithmetic (DA)-based

technique, has gained substantial popularity, in recent years, for their high-throughput processing capability, and increased regularity which results in cost-effective and area-time efficient computing structures. The main operations required for DA-based computation of inner-product are a sequence of look-up-table (LUT)-accesses followed by shift accumulation operations of the LUT output. DA-based computation is well-suited for FPGA realization, because the LUT as well as the shift-add operations can be efficiently mapped to the LUT-based FPGA logic structures.

In FIR filtering, one of the convolving sequences is derived from the input samples while the other sequence is derived from the fixed impulse response coefficients of the filter. This behavior of FIR filter makes it possible to use DA-based technique for memory-based realization. It yields faster output compared with the multiplier-accumulator-baseddesigns
because it stores the pre-computed partial results in the memory elements [8], which can be read out and accumulated to obtain the desired result. The memory requirement of DA-based implementation for FIR filters, however, increases exponentially with the filter order. DA was first introduced by Croisieret al [9]; and further developed by Peled and Lui [10] for efficient implementation of digital filters. Attempts are made to use offset-binary coding to reduce the ROM size by a factor of 2. An LUT-less adder-based DA approach has been suggested by Yoo and Anderson, where memory-space is reduced at the cost of additional adders. Memory-partitioning and multiple memory-bank approach along with flexible multi-bit data-access mechanisms are suggested for FIR filtering and inner-product computation in order to reduce the memory size of DA-based implementation. Allredetalhave suggested an efficient DA-based implementation of least mean square (LMS) adaptive filter using a decomposition of DA based FIR computation and subsequent memory decomposition. All these structures, however, are not suitable for implementation of the FIR filters in systolic hardware since the partial products available from the partitioned memory modules are summed together by a network of output adders. A new tool for the automatic generation of highly parallelized FIR filters based on PARO design methodology is presented, where the authors have performed hierarchical partitioning in order to balance the amount of local memory with external communication, and they have achieved higher throughput and smaller latencies by partial localization. A systolic decomposition technique is suggested in a recent paper for memory-efficient DA-based implementation of linear and circular convolutions. In this paper we have extended further the work to obtain an area-delay efficient implementation of FIR filter in FPGA platform.

PROBLEM STATEMENT

Due to the high performance requirements and increasing 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 termsof 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 and low area conscious design techniques in SoC include efforts at all level of abstraction. One way to efficiently incorporate high performance design technique is to implement IP cores

[4]. 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 theyrequire high order. The high order demands more hardware and area 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 delay. This is 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) Algorithmcan be used to replace MAC unit. The DA Algorithm actually uses lookup tablefor 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 and more speed. 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.

II. INTRODUCTION OF DISTRIBUTED ALGORITHM

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 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 telecommunications (e.g. Telecomm in Biomedical Signal Processing Communication, Wireless satellite and Image processing) which are performed efficiently. The multipliers in MAC unit of many DSP functions have more 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].

III.IMPLEMENTEDDA-BASED FIR FILTER FOR ASIC

The implemented structure of the DA-based FIR filter for ASICimplementation is shown in Fig. 1.

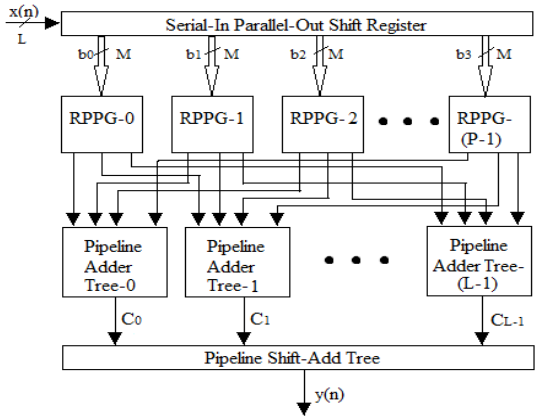


Fig. 1. Implementedstructure of DA-based FIR filters for ASIC.

The input samples $\{x(n)\}$arriving at every sampling instant are fed to a serial-in–paralleloutshift register (SIPOSR) of size$N$. The SIPOSR decomposesthe N recent most samples to P vectors $\mathbf{b}p$of length M for$p$ = 0, 1, . . . , P − 1 and feeds them to P reconfigurable partialproduct generators (RPPGs) to calculate the partial products.The structure of the implemented RPPG is depictedin Fig. 2 for M = 2.

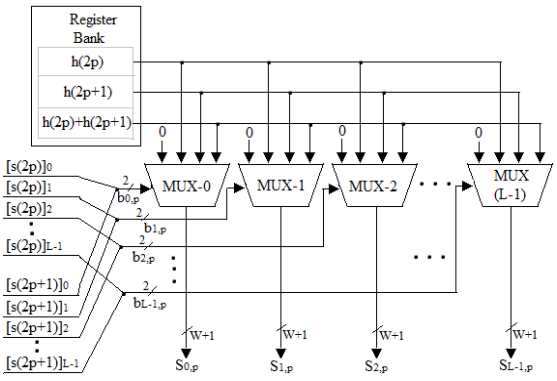


Fig. 2.pth RPPG for$M$= 2.

For high-throughput implementation, theRPPG generates L partial products corresponding to L bitslices in parallel using the LUT composed of a single registerbank of $2M$ − 1 registers and L number of 2M:1MUXes. In the implemented structure, we reduce the storage consumptionby sharing each LUT across L bit slices. The register array ispreferred for this purpose rather than memory-based LUT inorder to access the LUT contents simultaneously. In addition,the contents in the register-based LUT can be updated in parallelin fewer cycles than the memory-based LUT to implementdesired FIR filter. The width of each register in the LUT is(W + _log2M_) bits, where W is the wordlength of the filtercoefficient. The input of the MUXes are 0, $h(2p)$, $h(2p + 1)$,and $h(2p) + h(2p + 1)$; and the two-bit digit $\mathbf{b}l,p$is fed toMUX l for $0 \le l \le L - 1$ as a control word. We can find that$\mathrm{MUX}l$provides the partial product $Sl,p$for $0 \le l \le L$.

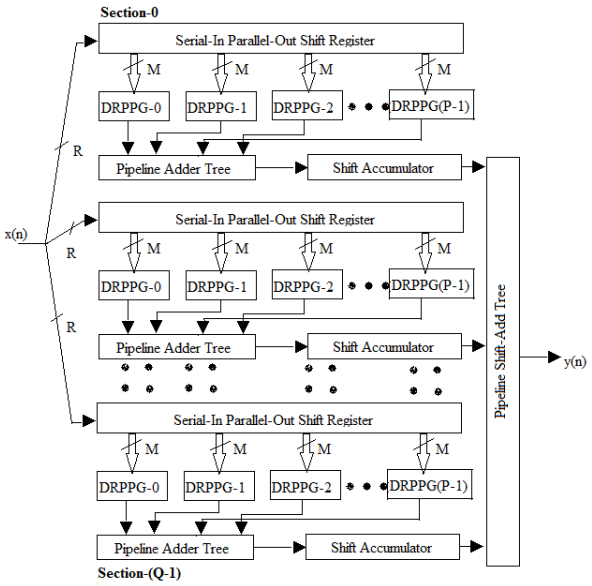
IV.IMPLEMENTEDDA-BASED FIR FILTER FOR FPGA

FPGA technology has tremendously grown from a dedicatedhardware to a heterogeneous system, which is considered to bea popular choice in communication base stations instead of beingjust a prototype platform. The implemented reconfigurable FIRfilter may be also implemented as part for the complete systemon FPGA. Therefore, here we implement a reconfigurable DAbasedFIR filter for FPGA implementation. The architecturesuggested in Section IV for high-throughput implementationof DA-based FIR filter is not suitable for FPGA implementation.The structure in Fig. 1 involves $N(2M - 1)/M$number of registers for the implementation of LUTs for FIRfilter of length N. However, registers are scarce resourcein FPGA since each LUT in many FPGA devices containsonly two bits of registers. Therefore, the LUTs are

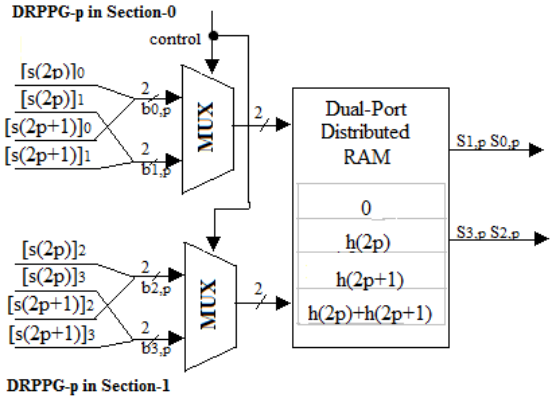
requiredto be implemented by distributed RAM (DRAM) for FPGAimplementation. However, unlike the case of the RPPG inFig. 2, the multiple number of partial inner products *Sl,p*cannot be retrieved from the DRAM simultaneously sinceonly one LUT value can be read from the DRAM per cycle.Moreover, if L is the bit width of input, the duration of thesample period of the design is L times the operating clockperiod, which may not be suitable for the application requiringhigh throughput. Using a DRAM to implement LUT for each bit slice will lead to very high resource consumption.Thus, we decompose the partial inner-product generator into$Q$ parallel sections and each section has R time-multiplexedoperations corresponding to R bit slices. When L is a compositenumber given by $L = RQ$ (R and Q are two positiveintegers), the index l can be mapped into$(r + qR)$ for $r = 0, 1, \ldots, R-1$ and $q = 0, 1, \ldots, Q-1$

$$y = \sum_{q=0}^{Q-1} 2^{-Rq} \left[\sum_{r=0}^{R-1} 2^{-r} \left(\sum_{p=0}^{P-1} S_{r+qR,\,p} \right) \right]$$

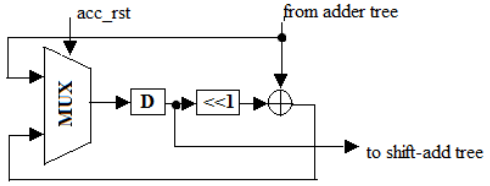
Fig. 3(a) shows the structure of the implemented time-multiplexed DA-based FIR filter using DRAM. To implement above equation, the structure has Q sections, andeach section consists of P DRAM-based RRPGs (DRPPGs)and the PAT to calculate the rightmost summation, followedby shift-accumulator that performs over R cycles according tothe second summation.



(a)



(b)



(c)

Fig.3.Implemented structure of the DA-based FIR filter for FPGA. (a) Structure of the DA-based FIR filter. (b) Structure of the DRPPG for$M$= 2and R = 2. (c) Structure of the shift-accumulator.

However, we can use dual-port DRAMto reduce the total size of LUTs by half since two DRPPGsfrom two different sections can share the single DRAM.

Thestructure of a DRPPG is shown in Fig. 3(b). The implemented structure can produce *QP* partial inner products in a singlecycle, whereas the structure in Fig. 1 can generate *LP* innerproducts. In the *r*th cycle, *P* DRPPGs in the *q*th section generate*P* partial inner products *Sr*+*qR*,*p*for *p* = 0, 1, . . . , *P* –1to be added by the PAT. The output of the PAT are accumulatedby a shift-accumulator [see Fig. 3(c)] over *R* cycles. Finally,the PSAT produces the filter output using the output from eachsection every *R* cycles. The accumulated value is reset every*R* cycles by the control signal [acc_rst in Fig. 3(c)] to keepthe accumulator register ready to be used for calculation ofthe next filter output. If the maximum operating clock periodis f_{clk}, the proposed structure can support the input samplerate of f_{clk}/R.

V.SIMULATION AND RESULTS

Implementation of FIR filter cores has been observed and we can see that fir filter cores have been implemented with both reference and DA structure. Results have been taken in terms of area utilized and speed performance for 16bits-20 taps and 8bits-20 taps. FIR filter cores have been designed in VerilogHDL and implemented using Xilinx 13.2 tool. Simulations were performed using Modelsim6.4b.

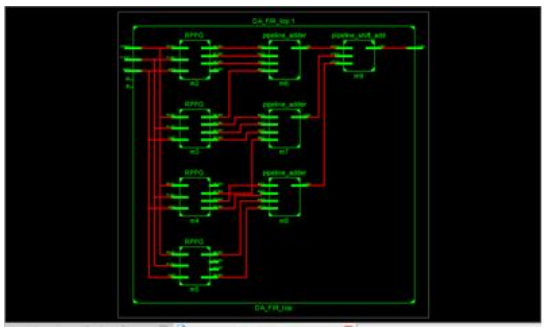


Fig.4.RTL schematic of FIR filter (ASIC)

Fig.5.Simulation results for ASIC

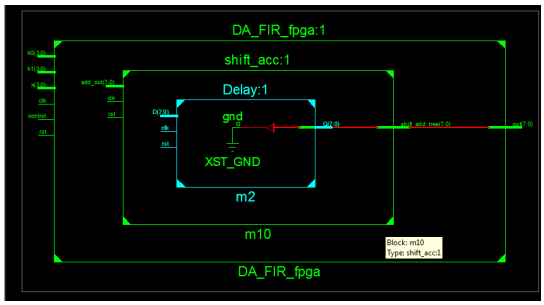


Fig.6.RTL schematic of FIR filter (FPGA)

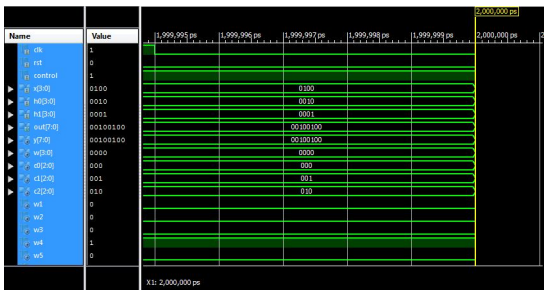


Fig.7.Simulation results for FPGA

Device Utilization Summary				
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Total Number Slice Registers	741	2,400	30%	
Number used as Flip Flops	101			
Number used as Latches	640			
Number of 4 input LUTs	967	2,400	40%	
Logic Distribution				
Number of occupied Slices	886	1,200	73%	
Number of Slices containing only related logic	886	886	100%	
Number of Slices containing unrelated logic	0	886	0%	
Total Number 4 input LUTs	1,022	2,400	42%	
Number used as logic	967			
Number used as a route-thru	55			
Number of bonded IOBs	57	92	61%	
IOB Flip Flops	16			
IOB Latches	32			
Number of GCLKs	1	4	25%	
Number of GCLKIOBs	1	4	25%	
Total equivalent gate count for design	13,610			
Additional JTAG gate count for IOBs	2,784			

Fig.8. Area Comparison (16 bit 16-taps)

Table 1 show that the area of Conv.UDF FIR Filter is less as compared

with same core implementedwith DA algorithm

Table 1. Area Comparison of 16 bit 16-taps

Filter Cores	Conv.UDF FIR Filter Core	Conv.UDF FIR Filter Core with DA Algorithm
No. of Slices	70%	74%
Slice Flip Flops	32%	37%
Input LUTs	38%	37%
Bonded IOBs	60%	60%
Total Eq Gate Count	13610	13573

Speed comparison (16 bit 16-taps)

Table 2 and Fig. 5 show results comparison of conv.UDF FIR Filter Core and conv.UDF FIR Core with DA Algorithm for speed. About 47% increase in speed has been found when using DA algorithm.

Table 2. Speed Comparison (16 bit 16-taps)

Filter Cores	Conv.UDF FIR Filter Core	Conv.UDF FIR Filter Core with DA Algorithm
Min Period	26.798 nS	18.161 nS
Input Arrival time	9.574 nS	9.574 nS
Output Req Time	15.842 nS	16.526 nS
Max Freq	37.316 Hz	55.063 Hz
Speed Improvement	_____	47.56 %

EXTENSION:

In this paper, we present an efficient architecture for the implementation of a delayed DA-based FIR filter. For achieving lower adaptation-delay andarea-delay efficient implementation, we use a novel partialproduct generator and propose a strategy for optimized balancedpipelining across the time-consuming combinational blocks ofthe structure, to reduce the adaptation delay as well.

Conclusion and Future Work

The results show that distributed arithmetic algorithm is better for FIR filters implementation on FPGAs. The efficiency in terms of area, speed has been analyzed. Comparison of results clearly shows that efficiency in terms of 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 thesis, 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 DAbased algorithm, serial distributed arithmetic algorithm and parallel distributed arithmetic use the look up table. The size of the look up table increases when the number of filter taps is increased. For better performance of FIR Filter, LUT lessDA implementation uses MUX. Every shift register uses MUX that select0 or filter coefficients, and this technique can be used to increase the efficiency of the FIR Filter in future.

REFERENCES

[1].T. Hentschel, M. Henker, and G. Fettweis, “The digital front-end of software radio terminals,” *IEEE Pers. Commun. Mag.*, vol. 6, no. 4, pp. 40–46,Aug. 1999.
[2] K.-H. Chen and T.-D.Chiueh, “A low-power digit-based reconfigurableFIR filter,” *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 53, no. 8,pp. 617–621, Aug. 2006.
[3] Xilinx, “Design Reuse Methodology for ASIC and FPGA Designers”, 2004. www.xilinx.com.
[4]. A.T.Erdogan, M.Hasan and T.Arslan, “Algorithmic low power FIR cores”, circuits and systems, IEEE proceedings, vol 150, pp. 155-169, 2003.
[5]. Heejongyoo and David V. Anderson, “Hardware efficient distributed arithmetic architecture for high order digital filters”, Acoustics, Speech, and Signal Processing,

2005. Proceedings. (ICASSP '05). IEEE, vol 5, pp. 125-128, 2005.
[6]. Wang sen, Tang Bin and Zhu Jun, "Distributed arithmetic for FIR filter design on FPGA", Communications, Circuits and Systems, 2007. ICCCAS 2007.IEEE, pp. 620-623, 2007.
[7]. P. K. Meher, "Hardware-efficient systolization of DA-based calculationof finite digital convolution," *IEEE Trans. Circuits Syst. II, Exp. Briefs*,vol. 53, no. 8, pp. 707–711, Aug. 2006.
[8]. Patrick Longa and Ali Miri, "Area efficient FIR filter design on FPGAs using distributed arithmetic", Symposium on signal processing and information technology, IEEE processing, pp. 248-252, 2006.
[9]. P. K. Meher, S. Chandrasekaran and A. Amira, "FPGA realization of FIR filters by efficient and flexible systemization using distributed arithmetic", IEEE Transactions on Signal Processing, vol. 56, pp. 3009-3017, 2008.
[10]. D. J. Allred, H. Yoo, V. Krishnan, W. Huang and D. V. Anderson, "LMS adaptive filters using distributed arithmetic for high throughput," IEEE Transactions on Circuits and Systems, vol, 52, pp, 1327-1337, 2005.