Volume 1, Issue 1
December 2012

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X

Design of Programmable, Efficient Finite Impulse Response Filter Based on Distributive Arithmetic Algorithm

Abdul Qayyum and Moona Mazher

Department of Computer Science, Federal Urdu University of Arts, Science and Technology, Islamabad, Pakistan
E-mail: qayyum_ce@yahoo.com

ABSTRACT

Present era of the mobile computing and multimedia technology demands high performance and low power Very Large Scale Integrated Circuit (VLSI) digital signal processing (DSP) systems. The availability of larger Field Programmable Gate Array (FPGA) devices has started a shift of System-on-Chip (SoC) designs towards using reprogrammable FPGAs, thereby starting a new era of System-on-a-reprogrammable-Chip (SoRC). One of the most widely used operations in DSP is Finite Impulse Response (FIR) filtering which performs the weighted summations of input sequences. Due to high speed requirements and increasing complexity of DSP systems, filtering operations have become computationally intensive and power expensive. This makes low power design an important area of research in field of digital design. The design of efficient FIR Filter in terms of low power, less area and high speed is the key issue in all signals processing application. Two designs are discussed in this thesis, one of them is Conventional Unfolded Direct Form FIR Filter core and the other one is the Conventional Unfolded Direct Form FIR Filter Core with distributed arithmetic algorithm. These two designs are compared with each other. In Distributed Arithmetic (DA) the task of summing product terms is replaced by table look-up procedures that are easy to implement in the Xilinx configurable logic block (CLB) look-up table architecture. Distributed Arithmetic (DA) plays a key role in embedding DSP functions in the Xilinx 4000 family of FPGA devices. The results show that the implementation of FIR Filter using DA Algorithm is more efficient as compared to FIR Filter Using Conventional Arithmetic Algorithm.

Keywords: *Finite impulse response, field programmable gate array, digital signal processing, distributed arithmetic, very large integrated circuits*

1. INTRODUCTION

In the electronic industry, there is a device known as Digital filters which are capable of taking digital input, process them and provide digital output. One of the basic types of digital filters is Finite Impulse Response (FIR) filter. These filters are 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 (DSP) 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

2. 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 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 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 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 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) Algorithm 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. 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^k, 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 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 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. BUILDING BLOCK OF FIR REFERENCE CORE

The existing core may be implemented using the main components which are given below:

- Counter
- Controller
- X sample value memory (X-RAM)
- B coefficient memory (B-Rom)
- Beta and gamma register
- Multiply-accumulator (MAC)
- Rounding
- Output sample (Y-Register)

The main components of the cores can be easily recognized from the top of module of the core. The block diagram of the existing direct form of FIR filter core is [8].

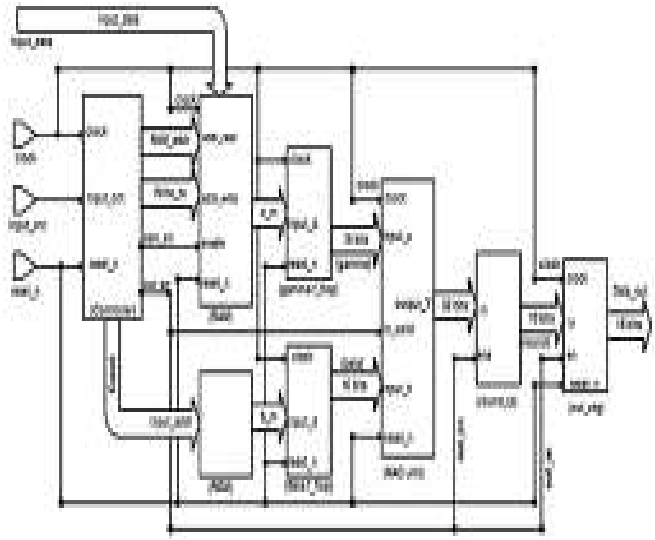

Fig.1. Block diagram of TOP module of the FIR Filter

5. PROPOSED MODEL

The basic derivation of DA has been proposed many authors in literature [10][11]. Let's start with the help of the basic equation of MAC unit which is given

$$y = \sum_{n=1}^{M} h_n x_n \qquad (1)$$

We can write x_n in K-bit Q1. (t-1) format [12], we have

$$x_n = -d_{n0}2^0 + \sum_{k=1}^{t-1} d_{nk}\,2^{-k} \quad (2)$$

We put this value of x_n in (1), we obtained

$$y = \sum_{n=1}^{M} h_n \left[-d_{n0} + \sum_{k=1}^{t-1} d_{nk}\,2^{-k}\right] \quad (3)$$

$$y = -\sum_{n=1}^{M} h_n d_{n0} + \left[\sum_{n=1}^{M}\sum_{k=1}^{t-1} h_n d_{nk}\,2^{-k}\right] \quad (4)$$

$$y = -\sum_{n=1}^{M} h_n d_{n0} + \sum_{k=1}^{t-1}\left[\sum_{n=1}^{M} h_n d_{nk}\right] 2^{-k} \quad (5)$$

$$S_0 = h_1 d_{10} + h_2 d_{20} + h_3 d_{30} + \cdots \ldots \ldots \ldots + h_M d_{M0} \quad (6)$$

$$S_k = h_1 d_{1k} + h_2 d_{2k} + h_3 d_{3k} + \cdots \ldots \ldots \ldots + h_M d_{Mk} \quad (7)$$

$$S_1 = h_1 d_{11} + h_2 d_{21} + h_3 d_{31} + \cdots \ldots \ldots \ldots + h_4 d_{41} \quad (8)$$

$$S_2 = h_1 d_{12} + h_2 d_{22} + h_3 d_{32} + \cdots \ldots \ldots \ldots + h_4 d_{42} \quad (9)$$

$$S_3 = h_1 d_{13} + h_2 d_{23} + h_3 d_{33} + \cdots \ldots \ldots \ldots + h_4 d_{43} \quad (10)$$

Using (7) and (8) in (6), we have

$$y = -[S_0] + \sum_{k=1}^{t-1} [S_K]\,2^{-k} \quad (11)$$

Expanding for t=4

$$\sum_{k=1}^{t-1} [S_K]\,2^{-k} = [S_1]2^{-1} + [S_2]2^{-2} + [S_3]2^{-3} \quad (12)$$

Using (13) into (12), we have

$$y = -[S_0] + [S_1]2^{-1} + [S_2]2^{-2} + [S_3]2^{-3} \quad (13)$$

The (14) can be written in reorder form and it is the final form of DA algorithm.

$$y = \{\{[S_3]2^{-1} + [S_2]\}2^{-1} + [S_1]\}2^{-1} - [S_0] \quad (14)$$

In these set of equations, the input data is the set of variable bits and coefficients of filters are constant (h_1, h_2, h_3, h_4). These constant coefficients are placed in a lookup-table. Input data are the address of the lookup table. The length of the filter coefficients and data will be the same. If four coefficients are used, then addresses of the lookup table will be 2^4. It means that increase in the length of the filter coefficients results an exponential increase in the size of the lookup table.

In Fig. 2 the data bits which are serially embedded in pipeline form make the addresses of the look up table. All filter coefficients are stored in a look up table as a partial product of the multiplicand which produces the final output.

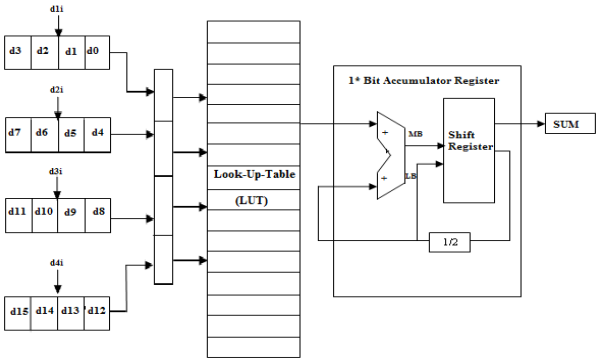

Fig.2.Block Diagram of Distributed Arithmetic Algorithm using look- up Table

6. IMPLEMENTING THE FIR FILTER USING 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.

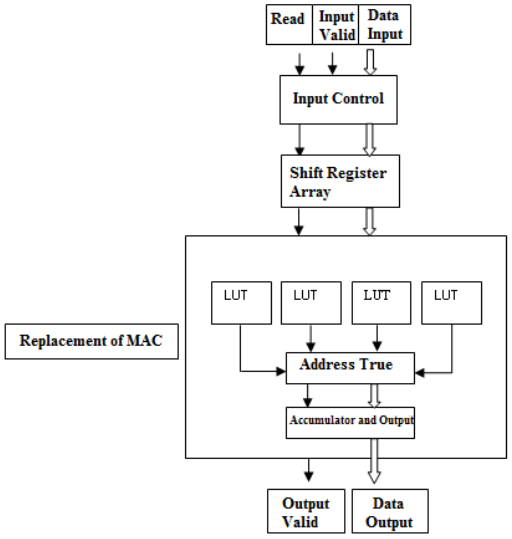

Fig.3.Implementation of FIR filter using DA

7. 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, power dissipated and speed performance for 16bits-20 taps and 8bits-20 taps. FIR filter cores have been designed in Verilog HDL and implemented using Xilinx 8.1i tool. Simulations were performed using Modelsim 5.7g.

The tool used for simulation is Xilinx 8.1 and results have been taken in area, speed and power. Furthermore the efficiency is also analyzed. The overall device utilization summary and performance analysis is given in Fig.4

FIR_REF Project Status			
Project File:	FIR_Ref.ise	Current State:	Placed and Routed
Module Name:	P_P_fir_core	• Errors:	No Errors
Target Device:	xc2s100-6tq144	• Warnings:	124 Warnings
Product Version:	ISE, 8.1i	• Updated:	Sun Dec 4 16:24:07 2011

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			

Performance Summary			
Final Timing Score:	0	Pinout Data:	Pinout Report
Routing Results:	All Signals Completely Routed	Clock Data:	Clock Report
Timing Constraints:	All Constraints Met		

Detailed Reports					
Report Name	Status	Generated	Errors	Warnings	Infos
Synthesis Report	Current	Sat Dec 3 23:48:10 2011	0	66 Warnings	3 Infos
Translation Report	Current	Sat Dec 3 23:48:14 2011	0	0	0

Fig.4. 16 bits data width and 16 bits taps with reference core

Design utilization summary of Conv.UDF Filter Core with DA is shown in Fig.5. Data input is 16 bit here with 16 taps used. Device utilization and performance summary shows the overall behavior of device with new DA algorithm after implemented on it.

FIR_FILTER_DA Project Status			
Project File:	FIR_Filter_DA.ise	Current State:	Placed and Routed
Module Name:	P_P_fir_core_with DA	• Errors:	No Errors
Target Device:	xc2s100-6tq144	• Warnings:	114 Warnings
Product Version:	ISE, 8.1i	• Updated:	Sun Dec 4 16:27:34 2011

Device Utilization Summary				
Logic Utilization	Used	Available	Utilization	Note(s)
Total Number Slice Registers	838	2,400	34%	
Number used as Flip Flops	198			
Number used as Latches	640			
Number of 4 input LUTs	866	2,400	36%	
Logic Distribution				
Number of occupied Slices	837	1,200	69%	
Number of Slices containing only related logic	837	837	100%	
Number of Slices containing unrelated logic	0	837	0%	
Total Number 4 input LUTs	868	2,400	36%	
Number used as logic	866			
Number used as a route-thru	2			
Number of bonded IOBs	57	92	61%	
IOB Flip Flops	16			
IOB Latches	32			
Number of GCLKs	2	4	50%	
Number of GCLKIOBs	2	4	50%	
Number of RPM macros	2			
Total equivalent gate count for design	13,573			
Additional JTAG gate count for IOBs	2,832			

Performance Summary			
Final Timing Score:	0	Pinout Data:	Pinout Report
Routing Results:	All Signals Completely Routed	Clock Data:	Clock Report
Timing Constraints:	All Constraints Met		

Detailed Reports					
Report Name	Status	Generated	Errors	Warnings	Infos
Synthesis Report	Current	Fri Nov 25 10:43:18 2011	0	67 Warnings	3 Infos

Fig.5.16 bits data width and 16 bits taps of reference core with DA

In this work, two FIR filter structures have been implemented completely. Efficiency of both designs has been measured. Comparison chart of both designs show that FIR filter with DA algorithm is more suitable for FPGAs than FIR filters with generalized structures. Area of both designs is almost same but efficiency enhancement has been found in speed and power analysis. Both 16 bit and 8 bit data with results have been explained having 16 numbers of taps.

Area Comparison (16 bit 16-taps)

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

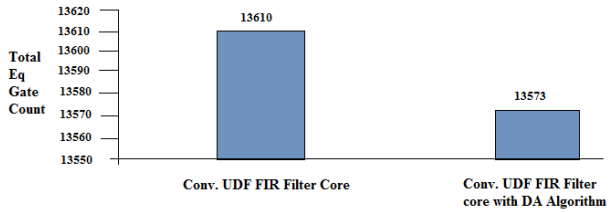

Fig.6. Area Comparison of 16 bit 16- taps

Speed comparison (16 bit 16-taps)

Table 2 and Fig. 7 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 %

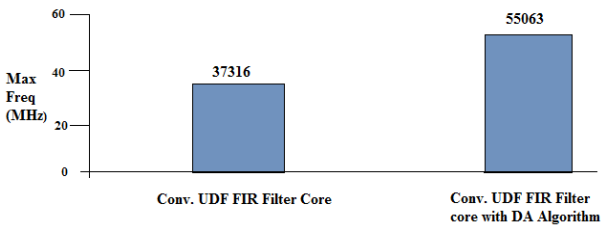

Fig.7. Speed Comparison of 16 bit 16-taps

Power comparison (16 bit 16-taps)

Figure 8 shows that power dissipation of FIR Filter with DA is less as compared with Conv.FIR Filter Core. There is about 21% increase in power efficiency.

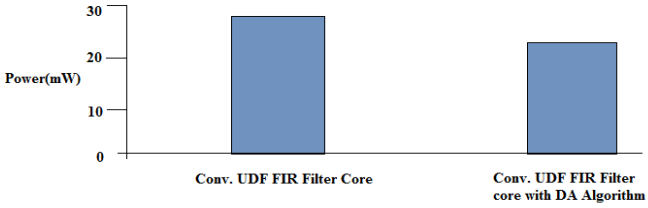

Fig.8. Power Comparison of 16 bit 16-taps

8. RESULTS 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 and power has been analyzed. Comparison of results clearly shows 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 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 DA based 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 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.

REFERENCES

[1]. Farhat Abbas Shah, Habibullah Jamal, Muhammad Akhter Khan, “Reconfigurable Low Power FIR Filter based on Partitioned Multipliers”, 16-19 December, KFUPM, Dhahran, KSA, ICM, 2006.

[2]. Xilinx, “Design Reuse Methodology for ASIC and FPGA Designers”, 2004. www.xilinx.com.

[3]. N. Sankarayya, K.Roy, D. Bhattacharya, "Algorithms for Low Power and High Speed FIR filter Realization Using Differential Coefficients", Analog and Digital Signal Processing, IEEE Transactions, vol, 44(6), pp. 488-497, 1997.

[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]. 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.

[8]. Muhammad Akhtar khan and A.T .Rrdogan, "Parameterized and programmable low power soft FIR filtering IP cores", Proceedings of the 4th WSEAS International Conference on Signal Processing, Computational Geometry & Artificial Vision, 2004.

[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.

AUTHOR PROFILES

Engr. Abdul Qayyum received his MS(Electronics Engineering) degree from IIU, Islamabad, Pakistan, in 2012. He is serving as Lecture in the department of Computer Science, Federal Urdu University of Arts, Science and Technology, Islamabad, Pakistan, since 2007. His research interests include Digital signal processing, Mobile Wireless and Multimedia Technology, FPGA, Digital systems and Speech and video Processing.

Moona Mazher received her BS(Telecom) degree from FUUAST, Islamabad, Pakistan, in 2009. She is serving as Lecture in the department of Computer Science, Federal Urdu University of Arts, Science and Technology, Islamabad, Pakistan, since 2010. Her research interests include Digital signal processing, Digital systems, Mobile Wireless and Image Processing and Artificial Intelligence.