FPGA based Efficient Interpolator design using DALUT Algorithm

Rajesh Mehra[1], Ravinder Kaur[2]

[1]Faculty of Electronics & Communication Engineering Department
[1]*rajeshmehra@yahoo.com*,
[2]ME Student of Electronics & Communication Engineering Department
[2]*rk_sid@yahoo.co.in*
National Institute of Technical Teachers'Training & Research,
Sector-26, Chandigarh, India

Abstract: Interpolator is an important sampling device used for multirate filtering to provide signal processing in wireless communication system. There are many applications in which sampling rate must be changed. Interpolators and decimators are utilized to increase or decrease the sampling rate. In this paper an efficient method has been presented to implement high speed and area efficient interpolator for wireless communication systems. A multiplier less technique is used which substitutes multiply-and-accumulate operations with look up table (LUT) accesses. Interpolator has been implemented using Partitioned distributed arithmetic look up table (DALUT) technique. This technique has been used to take an optimal advantage of embedded LUTs of the target FPGA. This method is useful to enhance the system performance in terms of speed and area. The proposed interpolator has been designed using half band poly phase FIR structure with Matlab, simulated with ISE, synthesized with Xilinx Synthesis Tools (XST) and implemented on Spartan-3E and Virtex2pro device. The proposed LUT based multiplier less approach has shown a maximum operating frequency of 92.859 MHz with Virtex Pro and 61.6 MHz with Spartan 3E by consuming considerably less resources to provide cost effective solution for wireless communication systems.

Keywords: MULTIRATE, FPGA, DALUT, FIR, LUT, MAC, XST

1 Introduction

The widespread use of digital representation of signals for transmission and storage has created challenges in the area of digital signal processing. Digital Signal Processing has become essential to the design and implementation of high performance audio, video, multi-media, and communication systems signal processing. An essential component of cost effective DSP algorithms is multirate signal processing.

Nabendu Chaki et al. (Eds.): NeTCoM 2010,CSCP 01, pp. 51–62, 2011.
 DOI : 10.5121/csit.2011.1105

The applications of digital FIR filter and up/down sampling techniques are found everywhere in modem electronic products. For every electronic product, lower circuit complexity is always an important design target since it reduces the cost. There are many applications where the sampling rate must be changed. Interpolators and decimators are utilized to increase or decrease the sampling rate. Up sampler and down sampler are used to change the sampling rate of digital signal in multi rate DSP systems [1]. This rate conversion requirement leads to production of undesired signals associated with aliasing and imaging errors. So some kind of filter should be placed to attenuate these errors

Today's consumer electronics such as cellular phones and other multi-media and wireless devices often require multirate digital signal processing (DSP) algorithms for several crucial operations in order to increase speed, reduce area and power consumption. Due to a growing demand for such complex DSP applications, high performance, low-cost Soc implementations of DSP algorithms are receiving increased attention among researchers and design engineers [2]. Although ASICs and DSP chips have been the traditional solution for high performance applications, now the technology and the market demands are looking for changes. On one hand, high development costs and time-to-market factors associated with ASICs can be prohibitive for certain applications while, on the other hand, programmable DSP processors can be unable to meet desired performance due to their sequential execution architecture. In this context, embedded FPGAs offer a very attractive solution that balance high flexibility, time-to-market, cost and performance. Therefore, in this paper, an interpolator is designed and implemented on FPGA device. An impulse response of an FIR filter may be expressed as:

$$Y = \sum_{k=1}^{K} C_k X_k \tag{1}$$

where C1,C2…….CK are fixed coefficients and the x1, x2…… xK are the input data words. A typical digital implementation will require *K* multiply-and-accumulate (MAC) operations, which are expensive to compute in hardware due to logic complexity, area usage, and throughput. Alternatively, the MAC operations may be replaced by a series of look-up-table (LUT) accesses and summations. Such an implementation of the filter is known as distributed arithmetic (DA).

2 Interpolator

In multirate systems, up sampler is basic sampling rate alteration device used to increase the sampling rate by an integer factor. An up sampler with an up-sampling factor *L*, where *L* is a positive integer, develops an output sequence $x_u[n]$ with a sampling rate that is *L* times larger than that of the input sequence $x[n]$. The up sampler is shown in Fig1

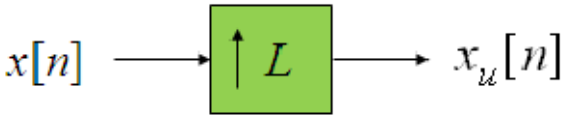

Figure1. Up Sampler

Up-sampling operation is implemented by inserting L-1 equidistant zero-valued samples between two consecutive samples of $x[n]$. The input and output relation of up sampler can be expressed as

$$X_u[n]=\begin{cases} x[n/L], & n = 0, \pm L, \pm 2L \ldots. \\ 0 & otherwise \end{cases} \quad (2)$$

The zero-valued samples inserted by the up-sampler are replaced with appropriate nonzero values using some type of filtering process called *interpolation* [3]. The input-output relation of an up-sampler with factor of 2 in the time-domain is given by:

$$x_u[n]=\begin{cases} x[n/2] & n=\pm 2,\pm 4\cdots \\ 0, & otherwise \end{cases} \quad (3)$$

The z transform of input output relation is given by

$$X_u(Z)=\sum_{n=-\infty}^{\infty} x_u[n]Z^{-n} = \left\{ \sum_{n=-\infty,}^{\infty} x[n/2]z^{-n} \right. \quad (4)$$

$$=\sum_{m=-\infty}^{\infty} x[m]z^{-2m} = X(Z^2) \quad (5)$$

In a similar manner, we can show that for a factor-of-L up-sampler

$$X_u(Z)=X(Z^L) \quad (6)$$

On the unit circle, for $z = e^{j\omega}$, input-output relation is given by

$$X_u(e^{j\omega})=X(e^{j\omega L}) \quad (7)$$

A factor-of-2 sampling rate expansion leads to a compression of $X(e^{j\omega})$ by a factor of 2 and a 2-fold repetition in the baseband $[0, 2\pi]$.This process is called imaging as we get an additional "image" of the input spectrum. Similarly in the case of a factor of-*L* sampling rate expansion,

there will be L-1 additional images of the input spectrum in the baseband. Interpolator is used as low pass filter to remove the xu[n] images and in effect "fills in" the zero-valued samples in $x_u[n]$ with interpolated sample values [4]-[6].

3 Distributed Arithmetic Algorithm

In the recent years, there has been a growing trend to implement digital signal processing functions in Field Programmable Gate Array (FPGA). Distributed Arithmetic (DA) appeared as a very efficient solution especially suited for LUT-based FPGA architectures. This technique, first proposed by Croisier is a multiplier-less architecture that is based on an efficient partition of the function in partial terms using 2's complement binary representation of data. The partial terms can be pre-computed and stored in LUTs. The flexibility of this algorithm on FPGAs permits everything from bit-serial implementations to pipelined or full-parallel versions of the scheme, which can greatly improve the design performance [7]

The multiplier less distributed arithmetic (DA)-based technique has gained substantial popularity, Due to its high-throughput processing capability and increased regularity, 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 lookup 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.[8]

Multiplier-less schemes can be classified in two categories according to how they manipulate the filter coefficients for the multiply operation. In first type of multiplier-less technique, the coefficients are transformed to other numeric representations whose hardware implementation or manipulation is more efficient than the traditional binary representation such as Canonic Sign Digit (CSD) method, in which coefficients are represented by a combination of powers of two in such a way that multiplication can be simply implemented with adder/subtractors and shifters [9] The second type of multiplier-less method involves the use of memories (RAMs, ROMs) or Look-Up Tables (LUTs) to store pre-computed values of coefficient operations.

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 the FIR filter makes it possible to use DA-based technique for memory-based realization. It yields faster output compared with the multiplier-accumulator-based designs because it stores the pre computed partial results in the memory elements, 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.

DISTRIBUTED ARITHMETIC (DA) is computation algorithm that performs multiplication with look-up table based schemes. DA specifically targets the sum of products (sometimes referred to as the vector dot product) computation that covers many of the important DSP filtering and frequency transforming functions. It uses look-up tables and accumulators instead of multipliers for computing inner products and has been widely used in many DSP applications such as DFT, DCT, convolution, and digital filters [10]. The example of direct DA inner-product generation is shown in Eq. (1) where ***xk*** is a 2's-complement binary number scaled such that |xk| < 1. We may express each xk as

$$x_k = -b_{k0} + \sum_{n=1}^{N-1} b_{kn} 2^{-n} \tag{8}$$

where the b_{kn} are the bits, 0 or 1, b_{k0} is the sign bit. Now combining Eq. (1) and (8) in order to express *y* in terms of the bits of x_k; we see

$$Y = \sum_{k=1}^{K} C_k [-b_k + \sum_{n=1}^{N-1} b_{kn} 2^{-n}] \tag{9}$$

The above Eq. (9) is the conventional form of expressing the inner product. Interchanging the order of the summations, gives us:

$$Y = \sum_{n=1}^{N-1} [\sum_{k=1}^{K} C_k b_{kn}] 2^{-n} + \sum_{k=1}^{k} c_k (-b_{k0}) \tag{10}$$

Eq.(10) shows a DA computation where the bracketed term is given by

$$\sum_{k=1}^{K} C_k b_{kn} \tag{11}$$

Each b_{kn} can have values of 0 and 1 so Eq.(11) can have 2K possible values. Rather than computing these values on line, we may pre-compute the values and store them in a ROM. The input data can be used to directly address the memory and the result. After *N* such cycles, the memory contains the result, *y*. The term x_k may be written as

$$X_k = ½\{x_k-(-x_k)\} \tag{12}$$

and in 2's-complement notation the negative of x_k may be written as:

$$-x_k = -\bar{b}_{k0} + \sum_{n=1}^{N-1} \bar{b}_{kn} 2^{-n} + 2^{-(N-1)} \tag{13}$$

where the over score symbol indicates the complement of a bit. By substituting Eq.(8) & (13) into Eq.(12), we get

$$x_k = \frac{1}{2}[-(b_{k0} - \bar{b}_{k0}) + \sum_{n=1}^{N-1} (b_{kn} - \bar{b}_{kn}) 2^{-n} - 2^{-(N-1)} \tag{14}$$

In order to simplify the notation later, it is convenient to define the new variables as

$$a_{kn} = b_{kn} - \bar{b}_{kn} \qquad \text{For } n \neq 0 \tag{15}$$

And

$$a_{k0} = b_{k0} - \bar{b}_{k0} \tag{16}$$

where the possible values of the a_{kn} **,** including n=0, are ± 1. Then Eq.(14) may be written as:

$$x_k = \frac{1}{2}[\sum_{n=0}^{N-1} a_{kn} 2^{-n} - 2^{-(N-1)}] \tag{17}$$

By substituting the value of x_k from Eq.(17) into Eq.(1), we obtain

$$Y = \frac{1}{2} \sum_{k=1}^{K} C_k [\sum_{n=0}^{N-1} a_{kn} 2^{-n} - 2^{-(N-1)}] \tag{18}$$

$$Y = \sum_{n=0}^{N-1} Q(b_n) 2^{-n} + 2^{-(N-1)} Q(0) \tag{19}$$

where

$$Q(b_n) = \sum_{k=1}^{K} \frac{C_k}{2a_{kn}} and Q(0) = \sum_{k=1}^{K} \frac{C_k}{2} \tag{20}$$

It may be seen that *Q(bn)* has only $2^{(K-1)}$ possible amplitude values with a sign that is given by the instantaneous combination of bits. The computation of ***y*** is obtained by using a $2^{(K-1)}$ word memory, a one-word initial condition register for *Q(O)* , and a single parallel adder sub tractor with the necessary control-logic gates.

4 Proposed Interpolator Design

Equiripple window based half band polyphase interpolator has been designed and implemented using Matlab [11]. The order of the proposed interpolator is 66 with interpolation factor of 2, transition width of 0.1 and stop band attenuation of 60 Db whose output is shown in Figure2.

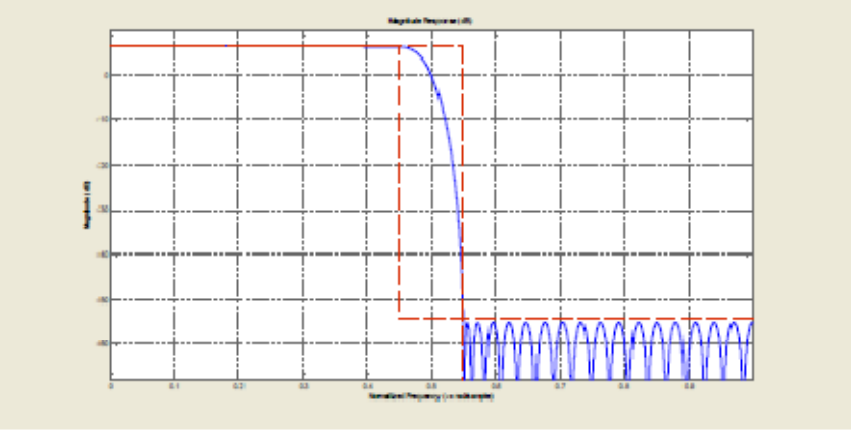

Figure2. Interpolator Response

Nyquist interpolators provide same stop band attenuation and transition width with a much lower order. In Half band filters about 50% of the coefficients of $h[n]$ are zero. This reduces the computational complexity of the proposed interpolator significantly.

4.1 Lth-Band Filters

Lth-band filters, of which the most popular is the half band (where L = 2), can be used to reduce hardware complexity because many of the coefficients are zero. When a coefficient is zero, the product of the multiplication is zero, so that particular multiplication may be omitted.
Half band filters are widely used in multirate signal processing applications when interpolating /decimating by a factor of two. Half band filters are implemented efficiently in polyphase form, because approximately half of its coefficients are equal to zero.

The transfer function of a half-band filter is thus given by

$$H(Z) = \alpha + Z^{-1} E_1(Z2) \quad (21)$$

with its impulse response is

$$h[2n] = \begin{cases} \alpha, & n = 0, \\ 0, & otherwise, \end{cases} \tag{22}$$

4.1 Design 1

The first interpolator design is has been implemented by using MAC based multiplier technique where 67 coefficients are processed with MAC unit as shown in Figure3.

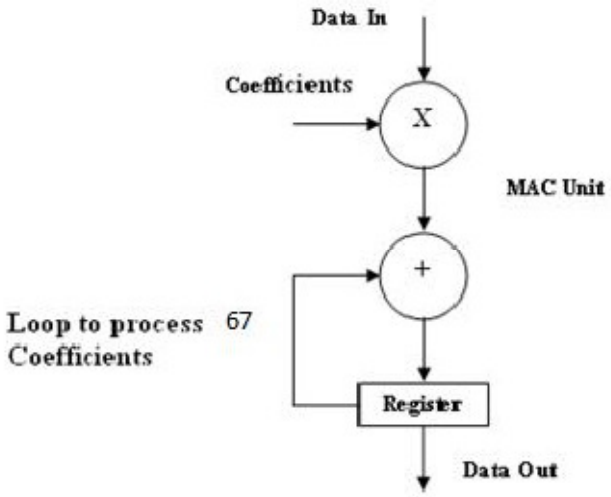

Figure3. MAC Based Multiplier Approach

4.2 Design 2

In the second interpolator design MAC unit has been replaced with LUT unit which is proposed multiplier less technique. Here 67 coefficients are divided in two parts by using polyphase decomposition. The proposed 2 branch polyphase interpolator structure is shown in Figure4 where interpolation takes place after polyphase decomposition to reduce the computational complexity and can be expressed as:

$$H(Z) = E_0(z^2) + Z^{-1}E_1(z^2) \tag{23}$$

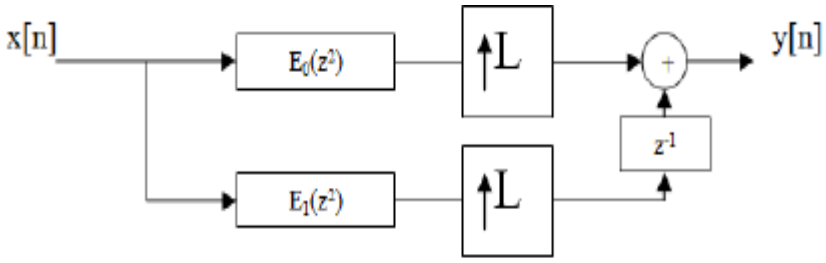

Figure4. Proposed Polyphase Interpolator

The coefficients corresponding to 2 branches $E_0(z2)$ and $E_1(z2)$ are processed by using partitioned distributed arithmetic look up table technique as shown in Figure5.

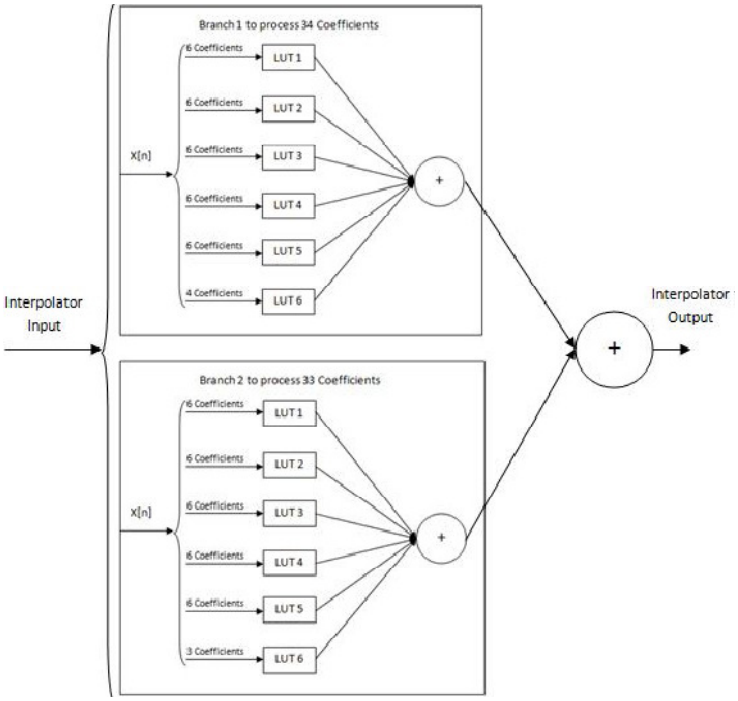

Figure5. Proposed LUT based Multiplier Less Approach

Each branch is processing the required coefficients using six partitions consisting of 6 LUTs. The two branches process the required coefficients in 6 6 6 6 6 4 and 6 6 6 6 6 3 manner respectively.

5 Hardware Simulation & Implementation

The MAC based and DA based interpolator designs have been synthesized and implemented on Spartan-3E based 3s500efg320-4 and Virtex 2 pro target device and simulated with ISE Simulator.

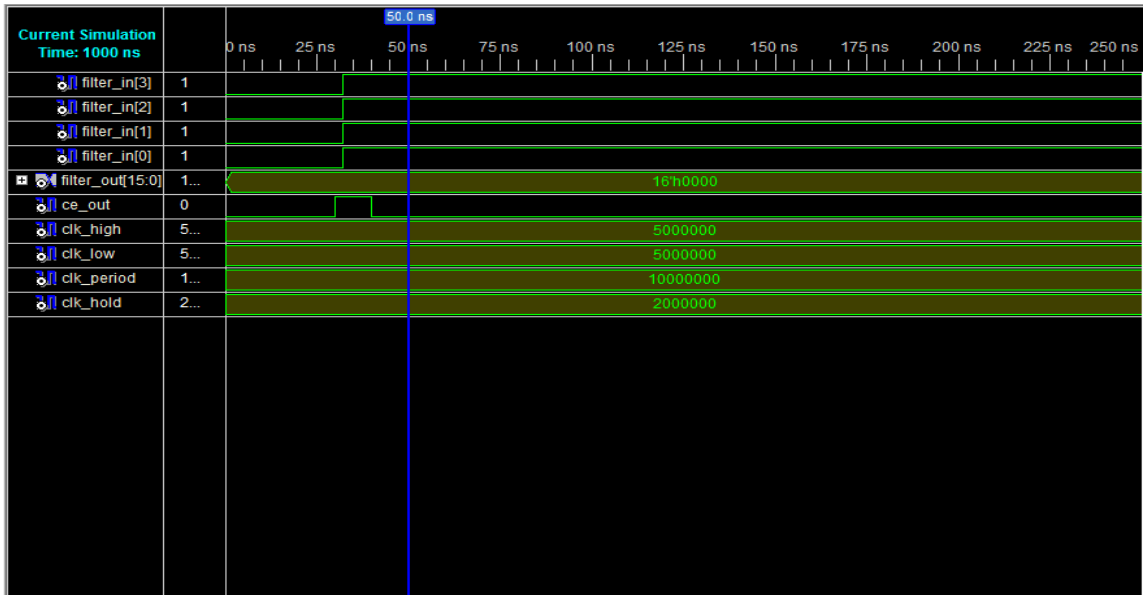

Figure6. Proposed Interpolator Response

TABLE 1 AREA & SPEED COMPARISON

Logic Utilization	Multiplier Approach		Multiplier less approach	
	SPARTAN 3E	VIRTEX2PRO	SPARTAN 3E	VIRTEX2PRO
# of Slices	590 out of 4656 (12%)	594 out of 13696 (4%)	309 out of 4656 (6%)	304 out of 13696 (2%)
# of Flip Flops	643 out of 9312 (6%)	644 out of 27392 (2%)	268 out of 9312 (2%)	249 out of 27392 (0%)
# of LUTs	568 out of 9312 (6%)	567 out of 27392 (2%)	485 out of 9312 (5%)	487 out of 27392 (1%)
#of Multipliers	1 out of 20(5%)	1 out of 136 (0%)	0 out of 20(0%)	-
Speed (MHz)	52.100	82.598	61.607	92.859

The ISE simulator based output response of proposed LUT based multiplierless interpolator with 16 bit input and output precision is shown in Figure6. The area and speed comparison of both techniques has been shown in table 1. The proposed LUT based multiplier less approach has shown a maximum operating frequency of 92.859 MHz with Virtex Pro and 61.6 MHz with Spartan 3E

The proposed multiplier less interpolator has consumed considerably less resources in terms of slices, flip flops and LUTs as compared to multiplier based design.

6 Conclusions

In this paper, an optimized equiripple based half band polyphase decomposition technique is presented to implement the proposed interpolator for wireless communication systems. The proposed interpolator has been designed using partitioned distributed arithmetic look up table approach to further enhance the speed and area utilization by taking optimal advantage of look up table structure of target FPGA. The proposed LUT based multiplier less approach has shown a a maximum operating frequency of 92.859 MHz with Virtex Pro and 61.6 MHz with Spartan 3E The proposed multiplier less interpolator has consumed considerably less resources in terms of slices, flip flops and LUTs and no multiplier of target device as compared to multiplier based design to provide cost effective solution for wireless and mobile communication systems.

7 References

[1]. ShyhJye Jou, Kai-Yuan Jheng*, Hsiao-Yun Chen and An-Yeu Wu, "Multiplierless Multirate Decimator/ Interpolator Module Generator", IEEE Asia-Pacific Conference on Advanced System Integrated Circuits, pp. 58-61, Aug-2004.

[2]. Vijay Sundararajan, Keshab K. Parhi, "Synthesis of Minimum-Area Folded Architectures for Rectangular Multidimensional", IEEE TRANSACTIONS ON SIGNAL PROCESSING, pp. 1954-1965, VOL. 51, NO. 7, JULY 2003.

[3] S K Mitra, Digital Signal Processing, Tata Mc Graw Hill, Third Edition, 2006.

[4] Ali AI-Haj, "An Efficient Configurable Hardware Implementation of Fundamental Multirate Filter Banks", 5th International Multi-Conference on Systems, Signals and Devices, pp.1-5, IEEE SSD 2008.

[5] Binming Luo, Yuanfu Zhao, and Zongmin Wang, "An Area-efficient Interpolator Applied in Audio Σ-DAC" Third International IEEE Conference on Signal-ImageTechnologies and Internet-Based System, pp.538-541, 2008.

[6] N.M.Zawawi, M.F.Ain, S.I.S.Hassan, M.A.Zakariya, C.Y.Hui and R.Hussin, "Implementing WCDMA Digital Up Converter In FPGA" IEEE INTERNATIONALRF AND MICROWAVE CONFERENCE, pp. 91-95, RFM-2008.

[7] Patrick Longa and Ali Miri "Area-Efficient FIR Filter Design on FPGAs using Distributed Arithmetic", pp248-252 IEEE International Symposium on Signal Processing and Information Technology,2006.

[8] Pramod Kumar Meher, , Shrutisagar Chandrasekaran, , and Abbes Amira," FPGA Realization of FIR Filters by Efficient and Flexible Systolization Using Distributed Arithmetic" IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 56, NO. 7, JULY 2008

[9] M. Yamada, and A. Nishihara, "High-Speed FIR Digital Filter with CSD Coefficients Implemented on FPGA", in Proc. IEEE Design Automation Conference (ASP-DAC 2001), 2001, pp. 7-8.

[10]. D.J. Allred, H. Yoo, V. Krishnan, W. Huang, and D. Anderson, "A Novel High Performance Distributed Arithmetic Adaptive Filter Implementation on an FPGA", in Proc. IEEE Int. Conference on Acoustics, Speech, and Signal Processing(ICASSP'04), Vol. 5, pp. 161-164, 2004

[11]. Mathworks, "Users Guide Filter Design Toolbox 4", March-2007.

Authors

Rajesh Mehra: *Mr. Rajesh Mehra is currently Assistant Professor at National Institute of Technical Teachers' Training & Research, Chandigarh, India. He is pursuing his PhD from Panjab University, Chandigarh, India. He has completed his M.E. from NITTTR, Chandigarh, India and B.Tech. from NIT, Jalandhar, India. Mr. Mehra has 14 years of academic experience. He has authored more than 30 research papers in national, international conferences and reputed journals. Mr. Mehra's interest areas are VLSI Design, Embedded System Design, Advanced Digital Signal Processing, Wireless & Mobile Communication and Digital System Design. Mr. Mehra is life member of ISTE*

Ravinder Kaur *is currently Senior Lecturer at Govt. polytechnic college,Punjab.She is pursuing her ME from NITTTR, Chandigarh, India and had done B.E from NIT, Srinagar, India.Ms Kaur has 23years professional experience. Ms Kaur interest areas are, Multirate Digital Signal Processing, Wireless & Mobile Communication and FPGA based embedded System Design*