Implementation of Digital Filters in Programmable Logic Devices based on Scale Free Hyperbolic CORDIC Processors

Shalini Rai[1], Rajeev Srivastava[2]

[12]Department of Electronics and Communication, University of Allahabad, Allahabad, India,

Abstract- There is a great replacement of the digital signal processing (DSP) processors with the programmable logic devices (PLDs) for the logic design engineer in very large scale integrated circuit (VLSI) technology with the minimization of chip area. The implementation of the different digital circuits in programmable logic devices (PLDs) is possible with the better performance, durability and stability of digital signal processing (DSP) processors. In recent era by using programmable logic density and hardware description languages (HDLs) the design of digital filters might be realizable in the VLSI technology with the proficiency of DSP processors. Here we explain the basic CORIDC algorithm, scale free CORDIC algorithm, programmable logic devices (PLDs) and low order digital filters. In this paper we design and give some experimental results of HDL synthesis of the first order digital filters whose designing is based on hyperbolic scale free CORDIC processor. These circuits are implemented on FPGA device (5vlx20tff323-2).The software Xilinx 13.1 and very high speed integrated circuit (VHSIC) hardware description language (HDL) VHDL language are incorporate in the designing of these filters.

Keywords- Scale free hyperbolic CORDIC processor, FIR Filter,IIR Filter, PLD's

I. INTRODUCTION

VLSI design engineers have two options one is the dedicated DSP processor or another one is either a programmable logic devices (PLDs) or application specific integrated circuit (ASIC) for the implementation of different digital circuits [1-3]. In this paper we embody the implementation of digital filters and programmable logic devices.PLD based implementation can have a lot of advantages over a dedicated processor. In integrated circuit (IC) technology for the hardware implementation design engineer try to improves the speed, cost as well as minimization of area to meet the moor's law, which says that in every ten years the component of IC becomes doubles. Programmable logic devices are important but hardware description languages (HDL) are also important for the implementing the different digital circuits like digital filters. HDLs are a language,which is acronym for hardware description language.It is text based digital circuit design where basic logic gates and registers are connected by different interconnection switches and also connected in schematic pattern. The HDLs which used by VLSI design engineers are Verilog and VHDL.The Verilog and VHDL are IEEE standards.These HDLs languages can be used for the designing of digital circuits in VLSI hardwrae. In this paper we use VHDL language for the implementation of first order digital filters.Here we describe the conventional CORDIC algorithm,and the advanced CORDIC algorithm i.e scale free CORDIC algorithm by which we are implemented first order digital filters. The FPGA device, here we used for the implementation of digital filters, is vertex-5(5vlx20tff323-2).

II. PROGRAMMABLE LOGIC FOR DIGITAL FILTERS

PLD's are also dedicated hardware as DSP procesoors.But they can achieve more efficient performance as compare to DSP processors. PLDs based processors consumes less power as compare to the DSP processors.The dedicated DSP processors are the more flexibile but they can require extra clock cycles as compared to the PLDs processor for the hardware implemention.The dedicated DSP processors are also power inefficient devices.So the logic designer prefer the PLDs processors.The embedded design engineer suggested that if the embedded system contains the programmable logic device on the system. The PLD may have the sufficient space available for a digital filters designing. If the logic space is not sufficient for the other complex circuitry, so the embedded design engineer, may be go with porting scheme i.e all the firmware ported with a dedicated DSP processors which one is suitable for the designing.

A. Basic's of Programmable Logic

PLDs are called programmable logic devices are used for the implementation and realization of logic functions in the VLSI hardware designing.The PLDs are made of configurable logic blocks,and they are interconnected by different fashion of interconnected switches,or predefined hardware logic resources such as gates and registers on integrated circuits. The combinational logic functions such as basic logic gates are easily implemented and for sequential logic functions, registers are commonly supported as flip –flops. Generally combinational logic functions are implemented on a small memory or look-up table (LUT) of Programmable logic Deices.A four input or five input combination function could be implemented in a RAM memory or look-up table. This is a four input Karnaugh mapping.For large complex combinational functions implemented using many LUTs which are distributed throughout the device, and LUTs can be connected by global routing channels.For the realization of sequential logic functions the look-up table can be drive by a flip –flop.The example of sequential logic functions are such as counter and state machine. Complex programmable logic devices (CPLDs) and Field Programmable Gate Arrays(FPGAs) are two subsections of the PLDs.CPLDs are combinations of many SPLDs (simple programmable logic Devices) and SPLDs are the combinations of PLAs (Programmable logic Arrays) and PALs (Programmable arrays Logic).CPLDs are based on non-volatile memory type technology and FPGAs are usually constructed using volatile memory similar to SRAM. CPLDs can be programmed one time and maintain their configuration permanently.But in FPGAs is temporarily programmed.FPGAs have more registers than CPLDs.FPGAs have more benefits than CPLDs for implementation of the different circuits.CPLDs are usually faster than FPGAs, having less propagation delay. The CPLDs are not available in large amount in market.

III. INTRODUCTION OF DIGITAL FILTERS

Finite impulse response and infinite impulse response filters are the two league of the digital filters [4,5].The finite impluse reponse filter is non-recursive filter i.e it has no feedback from output to the input and infinte impluse reponse filter is recursive filter i.e it has feedback from output to the input.The difference equations for both the digital filters are given in equations (1) and (2)

(1) $y_{\eta} = \sum \beta_k .x_{\eta-k} -\sum y_{\eta-k}. \alpha_k$ IIR Filter

(2) $y_{\eta} = \sum \alpha_k .x_{\eta-k}$ FIR Filter

Where α_k and β_k are the filter coefficients, integer η is the discrete-time increment, y_{η} the output of x_{η} ,the input of the filter. There are some basic differences between IIR and FIR filters as given below-

(1)FIR filters requires more filter coefficients or taps than IIR for same frequency response.(2) An FIR filter can be designed to have a linear phase response.(3)FIR filters can be configured in a symmetrical structure for saving logic resources area. Half number of multipliers are required for even order symmetrical FIR filters and it consumes less logic area. (4)An IIR filter contains poles. Stability is measured by checking that the magnitude of the roots of the denominator is less than unity, i.e. the poles are lies within the unit circle. (5) The analog filters are converted easily into IIR digital filter using standard transformations.By these techniques we can convert Butterworth, Bessel, and other conventional analog filters into digital IIR filters.

A. Design steps for the implementation of Digital Filters-

The good filter design there are some specific steps as given below(1) Specification of filter requirements (2) Computation of Filter coefficients (3) Presentation of appropriate structure of filter (4) Study of stability and finite word length effects (5) Calculation of the frequency response with the Z transforms (6) Implementation of the filter in hardware or software.

Digital Filter Structures.

There are three most important building blocks for the designing of the digital Filter structures.They are the module of delay, the summing and the multiplier.The combination of the multiplier and summing module is called as multiply and accumulate (MAC) module. The module of delay is called a filter tap. It is explained as recalling an old value from memory, storing the new value into memory after waiting t seconds, where t is the sampling period. The z transform of the module of delay (unit delay) is z^{-1} which can written as $e^{-j2\Pi ft}$= cos(2Πft)-sin(2Πft), where j^2= -1(Euler's identity).The summing module is block presentation of addition of the two signals.The multiplier is a block representation of multiplication of two signals or multiplication of a signal by a coefficient. These blocks are shown in Fig 1.These blocks are called signal flow diagram(SFD) or signal flow graph(SFG).

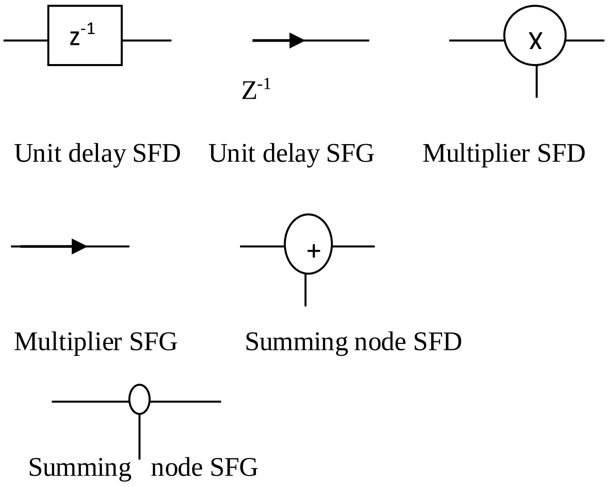

Fig.1:Graphical DSP building blocks

IV. FUNDAMENTAL OF CORDIC ALGORITHM

CORDIC [6,7] (Co-ordinate rotation digital Computer) algorithm is very useful algorithm for the computations of the linear, trigonometric and algebraic functions.This is a necessarry blocks for the calculations of the some important signals like sine and cosine which are important for the field of signal processing. The CORDIC algorithm is a vector rotation algorithm which is presented as follows:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} = \cos\theta \begin{bmatrix} 1 & \tan\theta \\ \tan\theta & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \qquad (1)$$

where (x',y') is the output vector resulting from rotation of input vector(x, y) through an angle θ.The angle θ is divided into several micro-rotations $\alpha_i = \tan^{-1}(2^{-i})$ of a set of iterative stages to rotate initial vector (x, y).The angle θ can be represents in terms of α_i as givem below

$$\theta = \sum_{i=0}^{m-1} d_i\alpha_i \qquad (2)$$

where i varies from 0 to m-1 and m is integer equal to bit precision and $d_i \in (+1, -1)$ depends upon the direction of rotations.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = k \begin{bmatrix} 1 & -d_i 2^{-i} \\ d_i 2^{-i} & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \qquad (3)$$

The value of d_i depends on the sign of the residue angle z_i where initial value of z_i is equal to $z_0 = \theta$. Residue angle during next iteration can be found using the micro rotation angle α_i as follows:

$$z_{i+1} = z_i - d_i\alpha_i \qquad (4)$$

The CORDIC algorithm is acted in three different coordinate system like circular, linear and hyperbolic. This theory [9,10] was suggested by the walther in 1971.He introduced one more variable in the iterative equations i.e μ as shown below

$$x_{i+1} = x_i - \mu d_i y_i 2^{-i}$$

$$y_{i+1} = y_i + \mu d_i x_i 2^{-i} \qquad (5)$$

$$z_{i+1} = z_i - d_i\alpha_i$$

as CORDIC used in circular, linear and hyperbolic trajectory respectively.

A. Scale free hyperbolic CORDIC Algorithm

The equations of the rotation matrix of unified CORDIC algorithm in hyperbolic coordinate system are mentioned below.

$$R_p = K_i \begin{bmatrix} 1 & \delta i.2^{-i} \\ \delta i.2^{-i} & 1 \end{bmatrix} \qquad (6a)$$

$$\theta_i = \tanh^{-1}(2^{-i}) \qquad (6b)$$

For conventional hyperbolic CORDIC processor the range of convergence (RoC) of angle is -1.118 to 1.118. The conventional hyperbolic CORDIC processor has some limitation of the range of convergence which restricts the hyperbolic processor at certain range of target angle. So VLSI design engineers design the hyperbolic CORDIC processor such as the iteration number i=4, 13, 40 executed twice and set the scale factor value is 1.207583. So the designer engineers want to the increment of range of target angle, elimination of scale factor and reduction of the number of iterations. So there are some needs of improvements in conventional CORDIC algorithm. A new methodology was proposed for the improvements of CORDIC algorithm. This is a scale free CORDIC algorithm [11-14], which are designed for increasing the range of convergence and reduction of the area consumption and number of iterations. Scale factor is completely abolished in scale free of CORDIC algorithm. The Coordinate equations expound in scale free CORDIC processor using taylor series. For finding the sequence of micro-rotation in scale free CORDIC algorithm use the most-significant-one bit location in the rotation angle. The bit presentation of the rotation angle is based on radix-2 format, and the micro-rotations are only in single direction, the Roc of the unified CORDIC algorithm is improved in scale free CORDIC algorithm by applying octant symmetry. The hyperbolic CORDIC rotation matrix is

$$Rp = \begin{bmatrix} \cosh\alpha i & \sinh\alpha i \\ \sinh\alpha i & \cosh\alpha i \end{bmatrix} \qquad (7)$$

The Taylor series expansion of the hyperbolic functions are given by

$$\cosh\alpha = 1 + \frac{\alpha^2}{!2} + \frac{\alpha^4}{!4} + --- \qquad (8)$$

$$\sinh\alpha = \alpha + \frac{\alpha^3}{!3} + \frac{\alpha^5}{!5} + --- \qquad (9)$$

The selection of the order of the taylor series in the scale free CORDIC algorithm depends on the accuracy, range of convergence and hardware complexity of the processor according the particular applications. The difference between calculated and approximated value of sinh and cosh is minimum for third order taylor series in the range of $[0, \pi/4]$.The percentage error in sinh and cosh values for third order approximation is also minimum as compare to the second order and fourth order taylor series approximation. So we select the third order taylor series for the rotation matrix in scale free hyperbolic CORDIC processor [15-17]. Therefore the rotation matrix of scale free CORDIC algorithm in hyperbolic mode is formulated as given below.

$$Rp = \begin{bmatrix} 1+\frac{\alpha i^2}{!2} & \alpha i+\frac{\alpha i^3}{!3} \\ \alpha i+\frac{\alpha i^3}{!3} & 1+\frac{\alpha i^2}{!2} \end{bmatrix} \qquad (10)$$

In the above mentioned rotation matrix replace factorial 3 to 2^2. So the rotation matrix becomes as:

$$Rp = \begin{bmatrix} 1+2^{-(2si+1)} & (2^{si}+2^{-(3si+2)} \\ (2^{si}+2^{-(3si+2)} & 1+2^{-(2si+1)} \end{bmatrix} \qquad (11)$$

These equations give the next coordinates values x_{i+1}, y_{i+1}.These equations depends on shift index values, shift index values calculated using the N word length and M which is the location of the most significant bit.

B. **Scale free hyperbolic CORDIC Processor**

In conventional CORDIC the elementary angles are defined previously and stored in a ROM. This conventional CORDIC algorithm is bidirectional i.e. the micro-rotation corresponding to all elementary angles rotated either clock-wise or anti clock wise, and each elementary angle is used only one time. But scale free CORDIC Processor is unidirectional, i.e. the micro-rotations are rotated in only one direction, and multiple iterations are occurred in scale free CORDIC algorithm. For accumulate elementary angles in scale free CORDIC processor[17], we reformulate the elementary angles as:

$$\alpha_i = 2^{-si} \qquad (12)$$

where s_i is the number of shifts for ith iteration. The most – significant- one bit location refers to the bit position of the leading one in an input string of bits starting from most significant bit (MSB).The MSO location identifier(MSO-LI) generates an n-bit output for a 2^nbit input string. It is used for finding the shift index

$$s_i = N - M \qquad (13)$$

N is the word length of the input data and M is the location of the most significant bit (one) in N input word length. The order of approximation of taylor series determines the largest fundamental angle of rotation. The basic shift and the largest fundamental angle for third order of approximation is to be:

$$s_{basic} = \left\lfloor \frac{b-log2(4!)}{4} \right\rfloor \qquad (14)$$

$$\alpha_{max} = 2^{-s_{basic}} \qquad (15)$$

where b is the word-length. For 16 bit word length, $s_{basic} = \lfloor 2.854 \rfloor$.Depending upon the desired accuracy, one can either select $s_{basic} = 2$ or s_{basic}=3. Any rotation angle θ is expressed as:

$$\theta = n_1.\alpha_{max} + n_2.\sum \alpha_{s_i} \qquad (16)$$

where $s_i \geq s_{basic}$ and n $= n_1 + n_2$,n is the total no of iterations ‘n’ is a constant. The third order taylor series gives the favorable results in only seven iterations. For designing of scale free CORDIC processor and micro-rotation sequence generator the input: angle is to be rotated θ_i.The most significant ones bit location is represented by MSO-LI (location identifier).If ML=15 then α=0.25 radians shift s_i = 2, θ_{i+1}=$\theta i - \alpha$. If one (1) bit location is other than 15 then shift index $s_i = 16 - ML \, and \;\; \theta_{i+1} = \theta_i \, with \, \theta_i[ML] = '0'$. **In Table I** shows the bit representation of elementary angles in decimal and hexadecimal according to the corresponding shifts. **Fig 2** shows the design of coordinate calculation unit for hyperbolic scale free CORDIC processor. The input coordinates are x_i , y_i and output coordinates are x_{i+1}, y_{i+1}.The coordinate calculation unit is designed using shift registers and adders. The shift index s_i calculation unit shows in **Fig 3.** For obtaining the next shift values in this unit we use shift registers and adders. The percentage error for the sinh and cosh value is undifferentiated for the range (0, π/4) in scale free CORDIC processor. So the maximum angle of rotation directed by micro-rotation sequence generation lies in the range (0,$\pi/4$), further for range improvement use octant symmetry property.

TABLE I
Angles and Respective Shifts

Shifts (si)	Elementary Angle(α_i)	
	Decimal	Hexadecimal
2	0.25	4000H
3	0.125	2000H
4	0.0625	1000H
5	0.0312	0800H

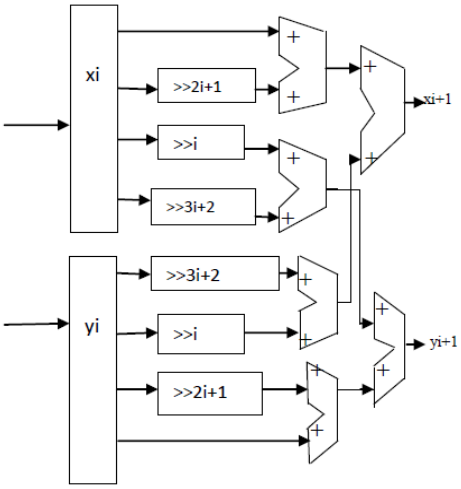

Fig.2: Co-ordinate calculation unit

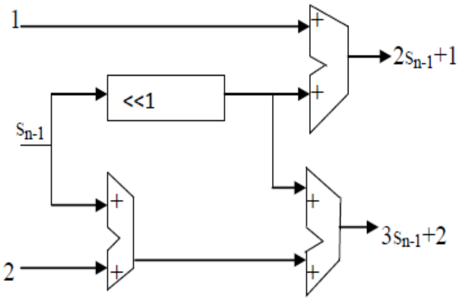

Fig.3: Shift Index calculation unit

C. Design of Digital Filter using Scale free hyperbolic CORDIC Processor

Here we presents the design of the low order digital filter (FIR and IIR) [18, 19] based on scale free hyperbolic CORDIC processor. Filter transfer function designing we use scale free hyperbolic CORDIC processor for the computation of the z^{-1} functions it means calculations of the unit delay, and for the multiplication of the filter coefficient values we use the convention linear CORDIC processor. So digital filter designing is based on the two main unit, first is the scale free hyperbolic CORDIC processor and second one is conventional linear CORDIC processor.

FIR Digital Filter-FIR filters are non-recursive filters**.** FIR Digital filters have integer valued impulse response coefficients. There are abundance of applications of the FIR filters in various engineering field because of their non-recursive, and linear phase characteristics.

Low Pass FIR Filter-The simplest low pass FIR filter is the moving average filter which has a transfer function

$$H_0(z)=\frac{1}{2}(1+z^{-1})=\frac{z+1}{2z} \qquad (17)$$

The above transfer function has a zero at z = -1 and a pole at z = 0.

High Pass FIR Filter-The simplest high pass FIR filter is obtained by replacing z with –z in low pass FIR filter. So resultant FIR high pass filter

$$H_1(z)=\frac{1}{2}(1-z^{-1})=\frac{z-1}{2z} \qquad (18)$$

The above transfer function has a zero at z=1 and a pole at z=0.

IIR Digital Filter – IIR filters are recursive filters**.** The IIR filters have infinite impulse response characteristics and non linear phase response. The analog filters can be converting easily into IIR filter. The standard transformations can be used to convert Butterworth, Bessel, and other conventional analog filters into digital IIR filters

Low Pass IIR Filter-A first order low pass IIR digital filter has a transfer function given by

$$H_{LP}(Z)=\frac{1-\alpha}{2}=\frac{1+Z^{-1}}{1-\alpha Z^{-1}} \qquad (19)$$

Where $|\alpha|<1$ for stability. The above transfer function has a zero at z= -1, i.e $\omega=\pi,$ which is in the stop-band of the filter. It has a real pole at z=α. As ω increases from 0 to π , the magnitude of the vector decreases from maximum value to 0. The maximum value of the magnitude of the function is unity at $\omega=0$, and the minimum value is zero at $\omega=\pi$.

$$\left|H_{LP}(e^{j0})\right|=1| \qquad H_{LP}(e^{j\pi})\right|=0$$

High Pass IIR Filter- A first order high –pass transfer function H $_{HP}$ (z) is given by

$$H_{HP}(z)=\frac{1+\alpha}{2}\frac{1-z^{-1}}{1-\alpha z^{-1}} \qquad (20)$$

Where $|\alpha|<1$ for stability. It has 3- dB cutoff frequency .The above transfer function has a zero at z= 1 i.e $\omega=0$. The maximum value of the magnitude of the function is unity at $\omega=\pi$, and the minimum value is zero at $\omega=0$

$$\left|H_{HP}(e^{j0})\right|=0 \qquad \left|\ H_{HP}(e^{j\pi})\right|=1$$

IV. RESULTS

(A) HDL Synthesis Report of FIR filter –

Device: Selected Device: 5vlx20tff323-2.

In **Table II, Table III, Table IV** we show the different parameters values in FIR filter circuit designing and implementation on FPGA device.

TABLE II

Parameters	**Number**
16 bit adders	45
16 bit registers	132
16 bit latches	35
16 bit comparators(<=)	280
17 bit comparators(<=)	7
16 bit (16 to 1) multiplexers	90
1 bit xor2s	2355
1 bit xor3s	2384

TABLE III

Parameters	**Numbers**	**%Utilization**
Slice Registers	780 out of 12480	6%
Slice LUTs	3054 out of 12480	23%
Bonded Input/Outputs	34 out of 172	19%
LUT Flip-Flop pairs	677 out of 3157	22%

TABLE IV

Parameters	**Value**
Time Period	39.052 secs
Frequency	25.607 MHz
Minimum input arrival time before clock	3.902 ns
Maximum output time after clock	49.357 ns
Combinational Path delay	0 ns

(B) HDL Synthesis Report of IIR filter –
Device: Selected Device: 5vlx20tff323-2.

In **Table V, Table VI, Table VII** we show the different parameters values in FIR filter circuit designing and implementation on FPGA device.

TABLE V

Parameters	**Number**
16 bit adders	46
16 bit registers	192
16 bit latches	35
16 bit comparators(<=)	280
17 bit comparators(<=)	7
16 bit (16 to 1) multiplexers	150
1 bit xor2s	4063
1 bit xor3s	3824

TABLE VI

Parameters	**Numbers**	**%Utilization**
Slice Registers	780 out of 12480	6%
Slice LUTs	3054 out of 12480	24%
Bonded Input/Outputs	34 out of 172	19%
LUT Flip-Flop pairs	677 out of 3157	23%

TABLE VII

Parameters	**Value**
Time Period	37.838secs
Frequency	26.429MHz
Minimum input arrival time before clock	3.902 ns
Maximum output time after clock	89.014ns
Combinational Path delay	0 ns

V. CONCLUSION

In this paper we explain the concepts of the basics of CORDIC algorithm and Programmable logic Devices, and how we implement the digital filters in programmable logic device using CORDIC algorithm. Here we use the scale free CORDIC algorithm by which we reduce the number of iterations of CORDIC processor and also reduce the area, power and delay in the designing of the hyperbolic CORDIC processor. By hyperbolic CORDIC processor we calculate the unit delay for the implementation of the digital filters. The linear CORDIC processor is used for the multiplication terms i.e multiplication of the signal to the coefficient value or signal to signal value. The frequency for scale free CORDIC algorithm based FIR filter circuit is 25.067 MHz, while frequency for the scale free CORDIC algorithm based IIR filter circuit is 26.429 MHz. The IIR filter circuit consumes more area than FIR filter circuit as mentioned above.

VI. REFERENCES

[1]. Farrell.J.D., Williams.D.M, "Implementation of Digital Filters in Programmable Logic Devices", class # .551 Embedded Systems Conference, San Francisco, spring 2001

[2]. FPGA and DSP Application Note, Xilinx Inc, 1996.

[3]. G.R.Goslin, "A Guide to Using FPGAs for Application-

Specific Digital Signal Processing”

[4]. Application Note, Xilinx Inc, 1995.

[5]. Andraka Raymond J, “A FIR Filter Fits in an FPGA Using a Bit Serial Approach”, 3rd Conference on Programmable logic Design.

[6]. Implementing FIR filters in Flex Devices, Application Note 73, Altera Corporation, February,1998.

[7]. J.E.Volder, “The CORDIC trigonometric computing technique,” IRE Transactions on Electronic Computers, vol.8, no.3 pp.330-334, 1959.

[8]. J.S.Walther, “A unified algorithm for elementary functions”, in proceedings of AFIPS Spring Joint Computer Conference, pp.379-385, May 1971.

[9]. Y.H.H., S.Naganathan, “An angle recoding method for CORDIC algorithm implementation,’’ IEEE Trans. Comput., vol42, pp.99-102, Jan.1993.

[10]. J.E.Volder, “The birth of CORDIC”, Journal of VLSI Signal Processing, Vol.25, no.2, pp.101- 105, 2000.

[11]. J.S.Walther, “The story of Unified CORDIC”, Journal of VLSI signal processing, vol.25, no.2, pp -107-112, 2000.

[12]. K.Maharatna et al., “Virtually scaling free adaptive CORDIC rotator,” IEEE Proc. Comp. Dig. Tech., vol.151, n0.6, pp.448-456, Nov.2004.

[13]. K.Maharatna et al., “Modified virtually scaling free adaptive CORDIC rotator algorithm and architecture,” IEEE Trans. Circuits Syst. Video Technol., vol.11, pp 1463-1474, Nov.2005.

[14]. P.K.Meher et al., “50 years of CORDIC: Algorithms, architectures and applications,” IEEE Trans. Circuits Syst.I, vol.56, no. 9, pp.1893-1907, Sep.2009.

[15]. F.Jaime et al., “Enhanced scaling free CORDIC,’’ IEEE Trans.Circiuts Syst.Video Technol., vol. 57, no.7, pp.1654-1662, Jul.2010.

[16]. Supriya Aggarwal, Kavita Khare, “Efficient Window – Architecture Design using Completely Scaling –Free CORDIC Pipeline”, 26th International Conference on VLSI Design and the 12 th International Conference on Embedded System, 2013.

[17]. Supriya Aggarwal and Kavita Khare, “Redesigned- Scale –Free CORDIC Algorithm Based FPGA Implementation of Window Functions to Minimize Area and Latency”. Hindwai Publishing Corporation, International Journal of Reconfigurable Computing Volume 2012, Article2012, Article ID 185784, 8 pages.

[18]. Supriya Aggarwal, Pramod Kumar Meher, Kavita Khare,“Scale-Free Hyperbolic CORDIC Processor and Its Application to Waveform Generation”, IEEE Transactions on Circuits and System-I Regular Papers, Vol.60, No.2, February 2013.

[19]. Sanjit K.Mitra, “Digital Signal Processing” A Computer-Based Approach Second Edition. ISBN No.0-07-044705-5.

[20]. Dr.K.V.V.K. Prasad, “Embedded/Real-Times System”, Concepts, Design & Programming. ISBN No. 978-81-7722-461-0.

Shalini Rai received the B.Tech, M.Tech degree in 2005 and 2008 respectively from Department of Electronics & Communication University of Allahabad, Allahabad. She has six years teaching experience. Currently she is research scholar of Department of Electronics and Communication, University of Allahabad. Her areas of interest are VLSI, Digital Electronics, Digital Signal Processing and Communication System etc.

Rajeev Srivastava received his B.Tech, M.Tech degree in Electronics Engineering in 1991, 1993 respectively from Department of Electronics and Communication University of Allahabad, J.K. Institute of Applied Physics & Technology, Allahabad. He has been teaching Electronic and Computer subjects at Allahabad University for more than twenty years. Currently he is Associate Professor in the Department of Electronics and Communication, University of Allahabad, Allahabad. His areas of interest are Computer Architecture, Microprocessor, Digital Electronics, Advanced Computer Architecture, DSP, Applied Numerical Analysis etc.