ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

Volume 3, Special Issue 3, March 2014

2014 International Conference on Innovations in Engineering (ICIET'14)

On 21st & 22nd March Organized by

K.L.N. College of Engineering, Madurai, Tamil Nadu, India

Area and Time Efficient FFT Architecture Using Hardwired Pre-Shifted Bi-Rotation Cordic Design

Manikandan .M[1], Paramasivam .C[2]

[1]Department of Electronics and Communication Engineering, K.S.R*angasamy* College of Technology,
Tirunchengode, India.

[2]Department of Electronics and Communication Engineering, K.S.R*angasamy* College of Technology,
Tirunchengode, India.

ABSTRACT-This paper presents CORDIC based feed forward FFT architecture. It is used to implement the pipeline FFT hardware architecture. The radix 2^kfeed forward FFT architecture can be used for the any number of parallel sample which is power of two. It can be achieved the high throughput and low hardware requirement. The hardwired pre shifted bi- rotation *cordic* technique for barrel-shifter of proposed circuit. Here two proposed CORDIC cells are used to the fixed angle rotations. This cells going to implement the micro rotations and scaling interleaved, it's implemented the two stages. The cascade proposed the bi-rotation CORDIC for higher throughput and reduced latency implementation. This method proposed optimized set of micro rotations for fixed and known angles. Shift and add circuits are used to implement the scaling factor. Fixed means square error used for analysis and reduced the error in this method. Synthesized the proposed CORDIC cells by Synopsys Design Compiler using TSMC 90-NM library, and shown that the proposed designs offer higher throughput, less latency and less area-delay product than the reference CORDIC design for fixed and known angles of rotation. We find similar results of synthesis of different Xilinx field-programmable gate-array platforms.

INDEX TERMS— FFT,Coordinate rotation digital computer (CORDIC), digital arithmetic, digital signal processing (DSP) chip, VLSI.

I. INTRODUCTION

Fast Fourier transform (FFT) is among the most widely used operations in digital signal processing. Often,a high performance FFT processor is the key component and determines most of the design metrics in many applications such as Orthogonal Frequency-Division Multiplexing (OFDM), Synthetic Aperture Radar (SAR) and software defined radio. For embedded systems, in particular portable devices; efficient hardware realization of FFT with small area, low-power dissipation and real-time computation is a significant challenge. A typical FFT processor is composed of butterfly calculation units, memory banks and control logic (address generator for data and twiddle factor accesses). In most cases, an FFT processor uses only one butterfly unit to realize all calculations iteratively, and the "in place" memory access strategy is required for the least amount of memory. With "in place" strategy, the outputs of a butterfly operation are stored back to the same memory location of the inputs, saving the memory usage by one half. However, correct memory addressing scheme is required to avoid the data conflict. This study implements an efficient addressing scheme to realize the parallel, pipelined and "in-place" memory accessing. It produces an output at every clock cycle; furthermore the memory banks and the butterfly unit are utilized with 100% efficiency within the pipeline. In FFT processors, butterfly operation is the most computationally demanding stage. Traditionally, a butterfly unit is composed of complex adders and multipliers, and the multiplier is usually the speed bottleneck in the pipeline of the FFT processor. The Coordinate Rotation Digital Computer (CORDIC) [5] algorithm is an alternative method to realize the butterfly operation without using any dedicated multiplier hardware. CORDIC algorithm is very versatile and hardware efficient since it requires only add and shift operations, making it very suitable for the butterfly operations in FFT [6]. Instead of storing actual twiddle factors in a ROM, the CORDIC-based FFT processor needs to store only the twiddle factor angles in a ROM for the butterfly operation. Additionally, the CORDIC-based butterfly can be twice faster than traditional multiplier based butterflies in VLSI implementations. The cordic based FFT architecture is used reduced hardware source and increase high throughput.

II. RADIX-2^2 FFT ALGORITHM

The -point DFT of an input sequenceX_n is defined as

$$X[K] = \sum_{n=0}^{N-1} X[n]W_N^{nk} \qquad k = 0,1,2 \dots\dots N-1$$

Where W_N^{nk} =$e^{-j(2\Pi/N)nk}$

When N is a power of two [15], the FFT based on the Cooley-Tukey algorithm is most commonly used in order to compute the DFT efficiently. The Cooley-Tukey algorithm

Reduces the number of operations from $o(N^2)$ for the DFT to $0(Nlog_2N)$ for the FFT. In accordance with this, the FFT is calculated in a series of n= log_pN stages, where is the base of the radix, of the FFT, i.e., Figs. 1 show the flow graphs of 16-point radix-2 and radix- FFTs, respectively, decomposed using decimation in frequency (DIF). At each stage of the graphs, butterflies and rotations have to be calculated.

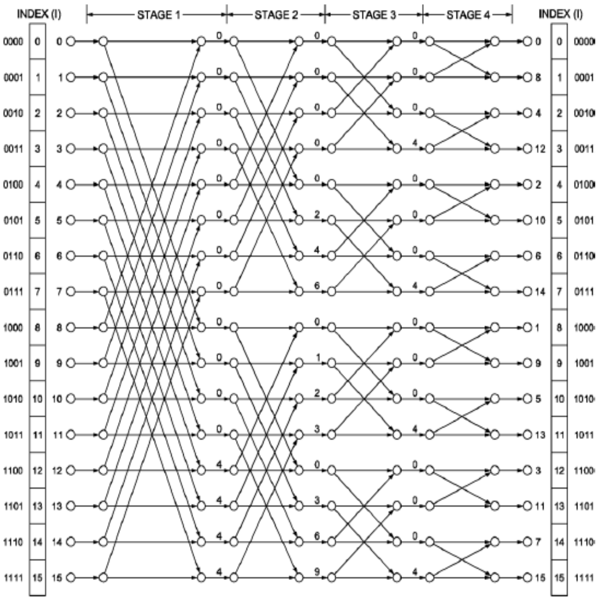


Fig 1. Flow graph of 16- point of Radix-2^2 DIF FFT

The lower edges of the butterflies are always multiplied by 1. These 1 are not depicted in order to simplify the graphs. The numbers at the input represent the index of the input sequence, whereas those at the output are the frequencies, of the output signal X[k]. Finally, each number, in between the stages indicates a rotation by

$$W_N^{\phi} = e^{-j2\Pi/n}\phi$$

As a consequence, samples for which $\phi = 0$ do not need to be rotated. Likewise, if $\phi = [0, N/4, N/2, 3N/4]$ the samples must be rotated by 0,270,180, and 90, which correspond to complex multiplications by 1,-j, j ,1 and respectively. These rotations are considered trivial, because they can be performed by interchanging the real and imaginary components and/or changing the sign of the data.Radix-2^2 is based on radix-2 and the flow graph DIF FFT can be obtained from the graph of a radix-2 DIF one.This can be done by breaking down each angle, at odd stagesinto a trivial rotation and a non-trivial one, where$\phi' = \phi$ mod N/4 and moving the latter to the following stage. Thisis possible thanks to the fact that in the radix-2 DIF FFT the rotationangles at the two inputs of every butterfly, and only differ by 0 or . Thus, if and, the rotation is moved to the following stage in accordancewithAnalogously, the radix- DIT FFT can be derived from the radix-2 DIT FFT. Contrary to DIF, for DIT the non-trivial rotations are moved to the previous stage instead of the following one.

III. HARDWIRED PRE SHIFTED BI-ROTATION CORDIC ALGORITHM

This technique is used to reduce the barrel shifter complexity. It has proposed the two methods,

1) Bi-Rotation CORDIC cell.
2) Case Cade Bi-rotation CORDIC.

Constants complex multiplication CORDIC circuit shown Fig 3. The control bits are stored in to the rom. SBR it is used to store the barrel –shifter and direction of micro-rotation. Micro rotations are implemented by the barrel shifter. The main contribution of the hardware is to implement the barrel-shifter and adder. Hard wired, pre shifting it's used to the minimization of barrel shifter complexity.

A. BI-ROTATION CORDIC CELL

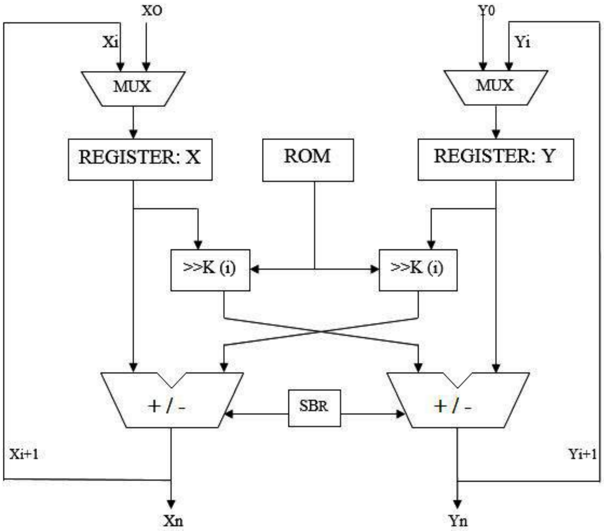


Fig 2. Constants complex multiplication CORDIC circuit

The proposed bi-rotation CORDIC circuit consists of an adder module, two multiplexer and sign bit register. The adder module consists of adder and subtracts which performs add and subtract. The output of the register sends in the two parts. One of the lines directly fed to the adder module. Another line fed into the barrel shifter. Pre-shifted by k (0) bit location shifted right by using hard wired pre shifting technique. The output of the loader is given in the register for the second CORDIC iteration. The barrel-shifters are 0 for the first- rotations and 1 for second micro rotations. Flip flop is used to generate the control bit and each cycle having the value 1 and 0.

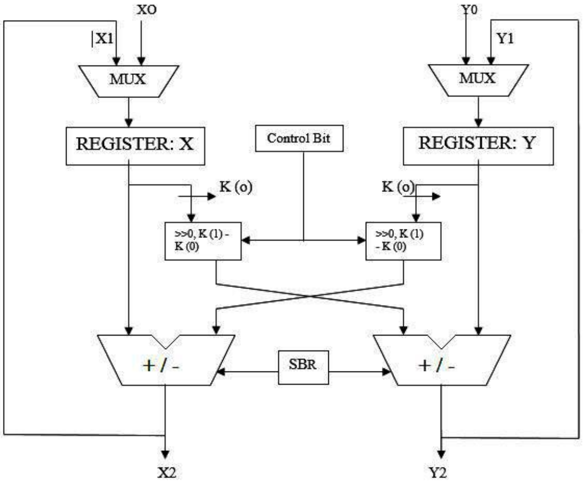


FIG 3. Hardwired pre-shifted bi-rotation. CORDIC

B. CASCADE BI-ROTATION CORDIC

Case Cade bi-rotation CORDIC it's used to improve the speed of the fixed angle rotation. It's designed by single and bi–rotation CORDIC. Case Cade bi –rotation CORDIC proposed the two cells follow as the

1) Case Cade CORDIC with single rotation cell
2) Case Cade CORDIC bi- rotation cell

The single rotation CORDIC cell has single rotation module Rotation module perform specific micro–rotations. Bi- rotation cell. It's is used to reduce the adder complexity of the single rotation CORDIC. Bi-rotation CORDIC used to implement the micro rotation. Here we have to proposed the two and three stage case code bi-rotation CORDIC for higher accuracy. The structure of the function it should be shown in the fig 4. The stage one four optimized micro rotations is implemented. While the rest two is performed by stage.

IV **PROPOSED CORDIC BASED FFT**

The proposed CORDIC based FFT is nothing but feed forward fft architecture.

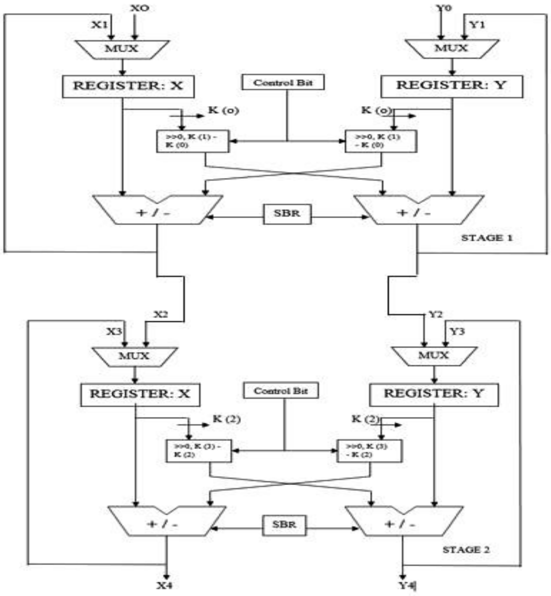


Fig 4 cascade bi- rotation CORDIC.

A. RADIX 2^2 FFT ARCHITECTURE

The proposed architectures have been derived using the framework presented in [15]. The design is based on analysing the flow graph of the FFT and extracting the properties of the algorithm. These properties are requirements that any hardware architecture calculates the algorithm must full fill. The properties of the radix- FFT are shown in Table I. The following paragraphs explain these properties and how they are obtained. The properties depend on the index of the data, I=$b_{n-1}, \dots . b0, b1$ where will be used throughout the paper to relate both the decimal and the binary representations of a number. This index is included both in decimal and in binary. On the one hand, the properties related to the butterfly indicate which samples must be operated together in the butterflies. This condition isb_{n-s} both for DIF and DIT decompositions and means that at any stage of the FFT, butterflies operate in pairs of data whose indices differ only in bit , where is the number$n = log_2 n$ of stages of the FFT. In it can be observed that at the third stage, data with indices. On the other hand, there are two properties for rotations. Atodd stages of the radix- DIF FFT only those samples whoseindex fulfils have to be rotated. These rotationsare trivial and the symbol indicates the logic ANDfunction. For the 16-point radix- FFT in only sampleswith indices 12, 13, 14, and 15 must be rotated at the first stage.For these indices is fulfilledb_{n-s} ,b_{n-s-1}, meeting the property, since and. Conversely, at evenstages rotations are non-trivial and they are calculated over indexed data for which where the symbolindicates the logic OR function.Analogously, the radix- DIT FFT can be derived from the

B. RADIX-2^2 FEED FORWARD FFT ARCHITECTURE

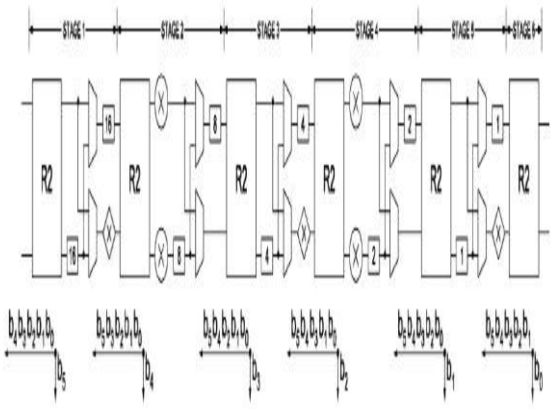


Fig 5-parallel radix-2^2 feedfoward FFT

The radix-2^2 feed forward architectures.First, a 16-point 4-parallel radix- feed forward FFT architectureis explained in depth in order to clarify the approach andshow how to analyse the architectures. Then, radix- feed forwardarchitectures for different number of parallel samples arepresented. Fig.2 shows a 16-point 4-parallel radix- feed forward FFTarchitecture. The architecture is made up of radix-2 butterflies(R2), non-trivial rotators, trivial rotators, which are diamond-shaped, and shuffling structures, which consist of buffersand multiplexers. The lengths of the buffers are indicated by anumber.The architecture processes four samples in parallel in a continuousflow. The order of the data at the different stages isshown at the bottom of the figure by their indices, together withthe bits that correspond to these indices. In the horizontal,indexed samples arrive at the same terminal at different time instants,whereas samples in the vertical arrive at the same timeat different terminals. Finally, samples flow from left to right.Thus, indexed samples (0, 8, 4, and 12) arrive in parallel at the inputs of the circuit at the first clock cycle, whereas indexed samples (12, 13, 14, and 15) arrive at consecutive clock cycles at the lower input terminal. Taking the previous considerations into account, the architecture can be analysed. First, it can be observed that butterflies always operate in pairs of samples whose indices differ in bit, meeting the property in Table I. For instance, the pairs of data that arrive at the upper butterfly of the first stage are: (0, 8), (1, 9), (2, 10), and (3, 11). The binary representation of these pairs of numbers only differ in. As, and at the first stage, so the condition is fulfilled.

This property can also be checked for the rest of the butterflies in a similar way.

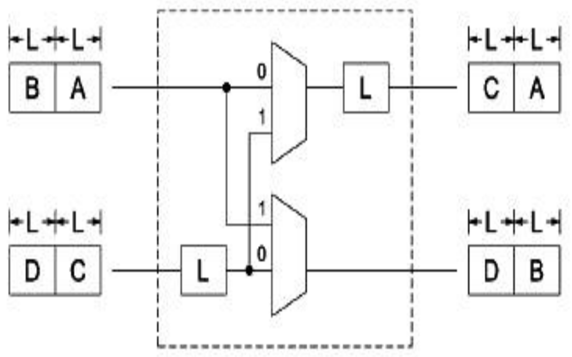


Fig 6. Circuit for data shuffling.

V. IMPLEMENTATION OF SCALING CIRCUIT

Scaling circuit, it's used to derived the shift add operations. Here CORDIC circuits realized the different level of implementation of micro–rotation [14].

$$T = \prod_{i=0}^{n-1}[1 + 2^{-2i}]^{-1/2}$$

A. SCALING FOR BI ROTATION CORDIC

Scaling and micro–rotation could be implemented in two separated pipeline stage. The scale factor in this case should be represented by two shifts – add terms

$$KA=1+\delta_0\, 2^{-s(0)} X\left(1 + \delta_1 2^{-s(1)}\right)$$

The two factor scaling can be implemented by the shift add circuit of hard wired pre shifting. It consists of adder and subtractors and a pair of single stage barrel shifter. It consists of the 2:1 mux. The input of the barrel-shifter is pre shifted by s (0) locations to right. The input through s(1)-s(0) location to right when the control bit is 1 or control bit is 0 means no additional shifts are required. The T flip-flop is generated the control bit.

B. SCALING FOR CASCADE BI ROTATION CORDIC

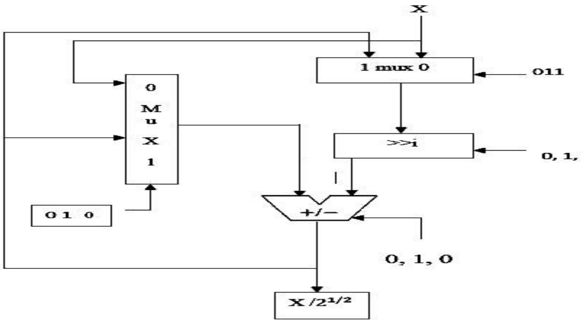


Fig 7. Scaling circuit for 45°and 43°

It is used to implement in two or three stages for four and six micro-rotations. The scaling circuit in 43 and 45 degree rotations through. We have proposed 43 and 45 degrees based on shown in below diagram.

TABLE I

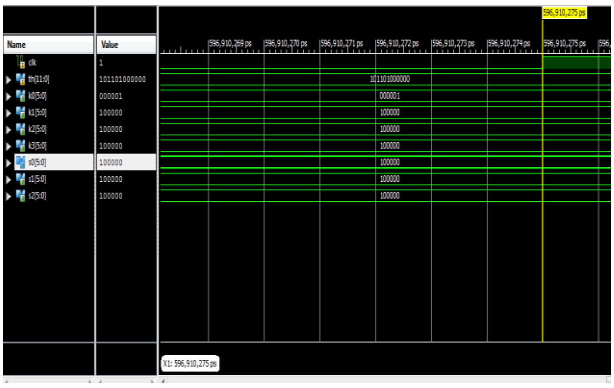
SYNTHESIS RESULT FOR AREA AND TIME EFFICIENT BI-ROTATION CORDIC

Designs	**Word size**	**Area (sq.um)**	**Clock**	**Tpt (Ms)**	**latency**	**ACT (ns)**	**ADP**
Reference Cordic circuit	16	2678	2.32	35.06	25.76	26.72	70220
	32	6565	4.15	13.88	71.08	72.08	473205
Hard wired pre shifted single rotation case cade cordic(1stage)	16	6560	1.41	542.46	12.36	1.71	11217
	32	23010	3.54	269.32	39.45	3.43	78924
Hard wired pre shifted Bi rotation case cade cordic(2stage)	16	5809	2.01	216.24	12.50	4.32	25094
	32	17667	2.96	116.26	42.56	6.92	122255
Cordic based feed forward fft architecture	16	4200	1.98	190.8	10.43	3.96	16,632
	32	15550	2.44	105.2	32.45	4.52	70,286

VI. RESULT AND CONCLUSION

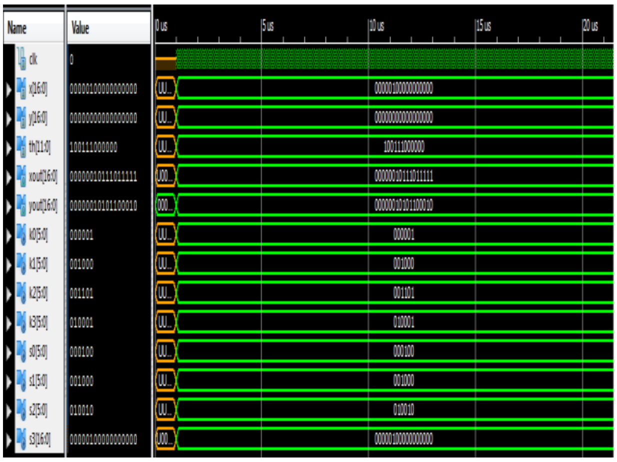
The proposed designs of the CORDIC based feed forward FFT architecture have been realized by VHDL. For basic CORDIC designs couldn’t find the any optimized specific anglesfor the vector-rotation and traditional hard wired, pre-shift bi-rotation CORDIC having the barrel shifter and adder complexities for fixed and known angle rotation. The proposed CORDIC based feed forward FFT architecture is used reduced the barrel and adder complexities. Improve the area and time for the proposed architecture.

The proposed CORDIC based feed forward FFT architecture require, respectively, 3.4 and 2.2 times more area over the reference design, but offer nearly 16.3 and 7.0 times more throughput, and involve nearly 4.6 and 2.5 times less ADP with nearly half and two-thirds of the latency of the other.The simulation is done in ModelSim and the code is functionally verified to be correct. Here TPT stands for throughput and calculated per second. ACT stands for average computation time measured in a nano second. ADP stands for area delay product, calculated as the product of the area and act in nano seconds.



Name	Value
clk	1
th[11:0]	101101000000
k0[5:0]	000001
k1[5:0]	100000
k2[5:0]	100000
k3[5:0]	100000
s0[5:0]	100000
s1[5:0]	100000
s2[5:0]	100000

Fig 8. Simulation result for ROM data in cascade bi - rotation CORDIC



Name	Value
clk	0
x[16:0]	00000100000000000
y[16:0]	00000000000000000
th[11:0]	100111000000
xout[16:0]	00000010111011111
yout[16:0]	00000010101100010
k0[5:0]	000001
k1[5:0]	001000
k2[5:0]	001101
k3[5:0]	010001
s0[5:0]	000100
s1[5:0]	001000
s2[5:0]	010010
s3[16:0]	00000100000000000

Fig 9.Simulation result of hard wired, pre shifted bi- rotation cascade CORDIC

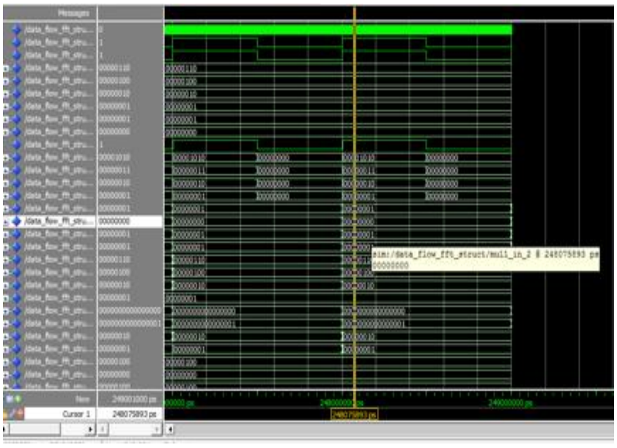


Fig 10. Cordic based feed forward FFT architecture

Fig 11. CORDIC based parallel radix-2^2 feedforward FFT

REFERENCE

[1] J. E. Volder, “The CORDIC trigonometric computing technique,” IRE Trans. Electron. Comput. Vol. EC-8, pp. 330–334, Sep. 1959.

[2] S.Y.Park and N. I. Cho, “Fixed-point error analysis of CORDIC processor based on the variance propagation formula,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 51, no. 3, pp. 573–584, Mar. 2004.

[3] T.-B. Juang, S.-F. Hsiao, and M.-Y. Tsai, “Para-CORDIC: Parallel CORDIC rotation algorithm,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 51, no. 8, pp. 1515–1524, Aug. 2004.

[4] T. Lang and E.Antelo, “High-throughput CORDIC-based geometry operations for 3D computer graphics,” IEEE Trans. Comput., vol. 54, no. 3, pp. 347–361, Mar. 2005.

[5] K. Maharatna, S. Banerjee, E. Grass, M. Krstic, and A. Troya, “Modified virtually scaling free adaptive CORDIC rotator algorithm and architecture,”IEEE Trans. Circuits Syst. for Video Technol., vol. 15, no. 11, pp. 1463–1474, Nov. 2005.

[6] C. Y. Kang and E. E. Swartz lander, Jr., “Digit-pipelined direct digital frequency synthesis based on differential CORDIC,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 53, no. 5, pp. 1035–1044, May 2006.

[7] C. H. Lin and A. Y. Wu, “Mixed-scaling-rotation CORDIC (MSRCORDIC) algorithm and architecture for high performance vector rotation *applications*,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 52, no. 11, pp. 2385–2396, Nov. 2005

[8] P. K. Meher, J. Valles, T.-B. Juang. Sridharan, and K. Maharatna, “*50 years of CORDIC: Algorithms, architectures and applications*,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 56, no. 9, pp. 1893–1907, Sep. 2009.

[9] K. Sridharan, and P. K. Meher, “*Efficient CORDIC algorithms and architectures for low area and high throughput implementation*,” IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 56, no. 1, pp.61–65, Jan. 2009.

[10] T.K.Rodrigues and E. E. Swartz lander, "*Adaptive CORDIC: Using parallel angle recoding to accelerate CORDIC rotations*," in Proc. 40th Asilomar Conf. Signals, Syst. Comput. (ACSSC), 2006.

[11] M.-P. Cani, T. Igarashi, and G.Wyvill, *Interactive Shape Design*. San Rafael, CA: Morgan & Claypool, 2007.

[12] Supriya Aggarwal**, "***Area-Time Efficient scaling-free CORDIC using generalized micro-rotation selection*". IEEE Transactions on Very Large Scale Integration (VLSI) Systems, AUGUST 2012.

[13] Alvaro Vazquez, "*Redundant Floating-Point decimal CORDIC algorithm*". IEEE TRANSACTIONS ON COMPUTERS, VOL. 61, NO. 11, NOVEMBER 2012.

[14] PramodkumarMeher. "*CORDIC Designs for Fixed Angle of Rotation*," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, February 2013.

[15] Mario Garrido. "Pipelined Radix- Feed forward FFT Architectures" IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, VOL. 21, NO. 1, JANUARY 2013.