IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
Volume 2, Issue 5 (May. – Jun. 2013), PP 09-16
e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197
www.iosrjournals.org

A New Memory Reduced Radix-4 CORDIC Processor For FFT Operation

Yasodai.A[1], Ramprasad.A.V[2]
[1]*Department Of ECE, Vickram College Of Engineering, Enathi, India-630561*
[2]*Professor, ECE, K.L.N College Of Engineering, Pottapalayam, India -630611*

Abstract: *A complex number can be interpreted as a vector in imaginary plane. The vector rotation in the x/y plane can be realized by rotating a vector through a series of elementary angles. These elementary angles are chosen such that the vector rotation through each of them may be approximated easily with a simple shift and add operation, and their algebraic sum approaches the required rotation angle. This can be exercised by CORDIC ('CO-ordinate Rotation Digital Computer) algorithm in rotation mode. In this paper, we have proposed a pipelined architecture new memory less z-path eliminated CORDIC algorithm for FFT computation. Pipelined architecture by pre computation of direction of micro rotation, radix-4 number representation, and the angle generator has been processed in terms of hardware complexity, iteration delay and memory reduction. Comparison of the proposed architecture with the conventional radix-4 architectures is elaborated. The proposed algorithm also exercises an addressing scheme and the associated angle generator logic in order to eliminate the ROM usage for bottling the twiddle factors. It incorporates parallelism and pipe line processing. The latency of the system is n/2 clock cycles. The throughput rate is one valid result per eight clock cycles. Additionally VLSI implementation on Virtex -4 FPGA is done. The implemented design operates at 450.654 MHZ of clock rate with a power consumption of 175.90mW.*

Keywords*-CORDIC, , FPGA, latency, Radix-4 , memory less systems, speed, throughput.*

I. Introduction

FFT is the heart of signal & image processing fields , also VLSI based signal processing find application in Neural Networks[2],bio-medical signal processing[4],wireless communications[7],Software Defined Radio[3], Radar image processing, Synthetic Aperture Radio.[6].. Most of the applications adopt CORDIC(Co-ordinate Rotation Digital Computer) based signal processing because of its versatility. Since then CORDIC algorithm was developed [8], it extends its application to new areas day by day. For FFT processors, butterfly operation is the most complicated stage. Generally butterfly unit consists of complex adders & multipliers usually multiplier is the speed bottleneck in the pipeline FFT processor [11]. CORDIC [8] algorithm is an iterative algorithm to realize the butterfly operation without using any dedicated multiplier hardware. CORDIC is an hardware efficient algorithm to realize transcendal functions, trigonometric functions, multiplications using only simple add & shift operations.

In recent years several CORDIC based FFT was proposed for various applications [1,2,3,4,5]. The conventional radix-2 CORDIC algorithm was implemented using word serial and digit pipelined architectures.[9] Later, this algorithm was modified as radix-4 to perform each micro rotation by two successive radix-2 micro rotations Lakshmi et al. in [9], presented a pipelined architecture using signed digit arithmetic for the VLSI efficient implementation of rotational radix-4 CORDIC algorithm, eliminating z-path completely. It achieves latency improvement with small hardware overhead. Garrido [14] proposed a new memory less CORDIC algorithm to realize FFT operation with reduced memory requirements But implementation is little complex. Vachhani. L et al. in [19] presented two area-efficient algorithms and their architectures based on CORDIC. Their first algorithm eliminated ROM and required only low-width barrel shifters and their second algorithm eliminated barrel shifters completely. As a consequence, both the algorithms consume approximately 50% area in comparison with other CORDIC designs but hardware complexity is large. In [16], a modified CORDIC algorithm for FFT processors was intended which wipes out the need for bottling the twiddle factor angles. The algorithm generates the angles successively by an accumulator. This approach reduces the memory requirements of an FFT processor by more than 20% [11]. Also, memory reduction improves with the increased radix size. Moreover, the angle generation circuit consumed less power consumption than angle memory accesses. Instead of storing actual twiddle factors in a ROM, CORDIC based FFT processor needs a dedicated memory bank to store actual twiddle factor angles for butterfly operation. In this paper we have proposed a reduced memory z- path eliminated radix -4 CORDIC for FFT operation It leads to reduced power consumption, compared to conventional CORDIC.

The organization of the paper is as follows: In section 2 Fundamentals of CORDIC algorithm and its operation in FFT is discussed. In section 3 proposed methodology of z-path eliminated memory efficient CORDIC algorithm is discussed. In section 4 synthesis results are discussed.

II. Radix-4-CORDIC-FFT algorithm

N point Discrete Fourier Transform is defined as

$$X(k) = \sum_{i=0}^{N-1} x(i) * W_N^{ik} \qquad (1)$$

where; k = 0,1,...,n – 1 and $W_N^{ik} = e^{-j\frac{2\pi}{N}ik}$ is the "twiddle factor". There will be $\log_r N$ stages and each stage contains N/r butterfly operations, in an N-point FFT. The FFT-butterfly operation can be realized by radix-r cordic algorithm. CORDIC algorithm was initially proposed by J.E.Volder 8[ ]. Because the CORDIC-based butterfly can be fast. The foundation for CORDIC algorithm is a 2D vector rotation in the xy-plane (see Fig.1). The vector rotation can be realized by rotating a vector through a series of elementary angles. These elementary angles are chosen such that the vector rotation through each of them may be approximated easily with a simple shift and add operations, and their algebraic sum approaches the required rotation angle [8]. The CORDIC algorithm can be exercised in two different modes, viz. Rotation mode and Vectoring mode.

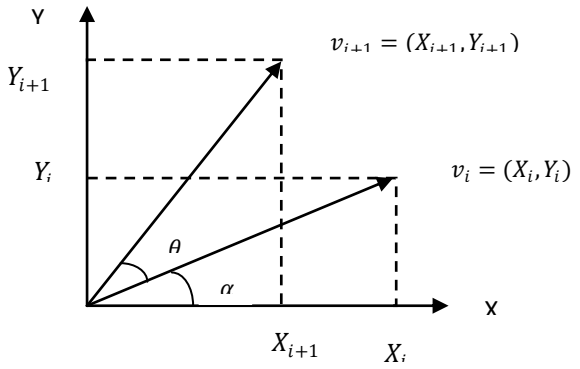

Fig (**1) A 2D vector rotation using CORDIC**

The rotation mode is committed to practice general rotation for the given angle and to compute elementary operations such as multiplication, exponential, trigonometric functions and hyperbolic functions counting on the coordinate system. The vectoring mode is utilized to define the angular argument of the original vector and estimate the logarithmic and division functions.The standard CORDIC system in circular coordinate system practices 'n' iterations for n-bit precision using radix-r number representation. For a given n-bit precision, the increase of radix-rcuts down the number of micro rotations. For example, the radix-4 CORDIC system achieves latency reduction which is an extension of radix-2 CORDIC algorithm.[9] Using radix-4 we achieve half the number of micro rotations than that required in radix-2. The iteration equations of the radix-4 CORDIC algorithm at the$(i+1)^{th}$ step [ ] under circular coordinate system acting in rotation mode are as follows:

$$X_{i+1} = X_i - \sigma_i * Y_i * 4^{-i} \qquad (2a)$$

$$Y_{i+1} = Y_i + \sigma_i * X_i * 4^{-i} \qquad (2b)$$

$$Z_{i+1} = Z_i - \tan^{-1}(\sigma_i * 4^{-i}) \qquad (2c)$$

Where, the direction of rotation in each iteration is $\sigma_i \epsilon \{-2, -1, 0, 1, 2\}$, and an elementary angle $\tan^{-1}(\sigma_i * 4i$. The realization of Zi+1 in Eq. (2c) increases the length of the vector. To up hold the expected value of the vector, $(i+1)^{th}$ coordinates must be multiplied by the scale factor. The scale factor depends on the values of σ_i and must be estimated for each rotation angle. $K^{-1} = \prod_{i\geq 0} K_i^{-1} = \prod_{i\geq 0} \sqrt[-1/2]{1+\sigma_i * 4^{-2i}}$ (3)

III. Proposed Methodology:

Radix -4 CORDIC for pipelined FFT operation, with σ prediction block is proposed to reduce iteration delay, eliminate the z-path and also to cut down the memory required to store the twiddle factor angle. The number of iterations is reduced by using radix-4 number system and iteration delay is reduced by radix-4 signed digit arithmetic and pre computation of direction of micro rotation. The radix-4 takes more time to select from among the five rotation direction values, and to select an appropriate angle out of five elementary angles, which increases the iteration delay compared to the radix-2. Nevertheless, the proposed CORDIC architecture in rotation mode is designed to overcome this issue by pre compiling all the direction f rotations for the generated

input angle utilizing the linear relation between the rotation angle and the framed binary representation of directions of micro rotations [9]. The proposed architecture comprises of four basic building blocks viz., the angle generator, the x/y path, the σ-prediction block, and the scale factor computation blocks shown in Fig.2 The angle generator block produces the sequence of angles in regular and increasing order with the help of an accumulator and control logic. Theσ-prediction block predicts all the σ_i values in radix-4 interpretation for the generated target angleθ, prior to the true CORDIC rotation starts iteratively in the x/y path. During every clock cycle, the σ_i values are buffered and sent sequentially to the corresponding stages of x/y path.

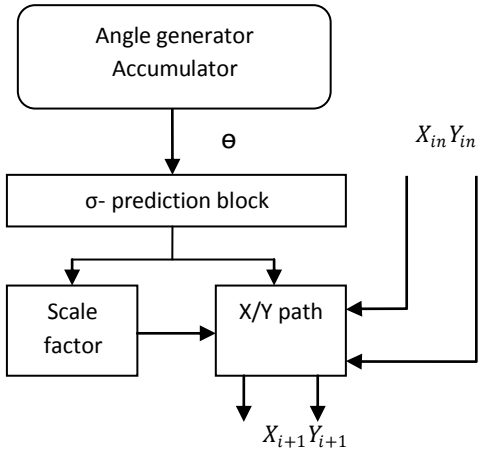

Fig 2 Block schematic of the proposed system

1) θ-(Angle) generator block:-

The key operation of FFT is as shown in Eq. (1), $X(k) = \sum_{i=0}^{N-1} x(i) * W_N^{ik}$ .This is equivalent to Rotate x(i) byan angle$(-j\frac{2\pi}{N}ik)$, which can be realized very well by the CORDIC algorithm with simple shift and add operations, CORDIC-based butterfly can be fast. As we know that, an FFT processor needs to store the twiddle factors in memory. But, the CORDIC-based FFT doesn't have twiddle factors but needs a memory bank to store the rotation angles.

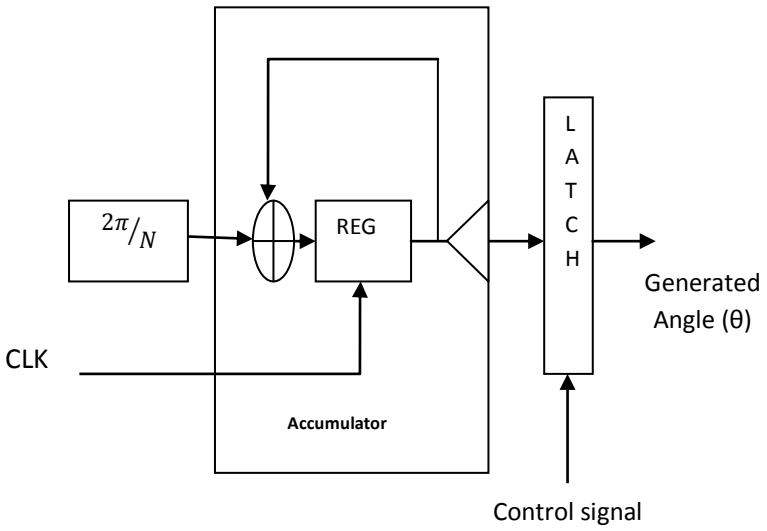

Fig 3 θ- generator of the proposed system.

Using a addressing scheme presented in [15], the twiddle factor angles generated by a simple accumulator, follow a regular and increasing order. For example 64 -point radix-4 FFT, it can be seen that twiddle factor angles are sequentially increasing, and every angle is a multiple of the basic angle$(\frac{2\pi}{N})$, which is$(\frac{\pi}{32})$. For $(\log_4 64) = 3$ different FFT stages, the angles increase always one step per clock cycle. This function can be realized by the angle generator circuit which is composed of an accumulator, and an output latch, as shown in Figure 3. The accumulator output depends on the Control signals those enable or disable the latch which is simple to realize, as it is based on the present FFT butterfly stage and RAM address bits

$b_n \ldots b_2 b_1 b_0$ (see Table 1). Usually, for an N = 2^n point FFT, an $(n-1)$ bit butterfly counter-$B(b_{n-2} b_{n-3} \ldots b_1 b_0)$will satisfy the address sequences and the control logic of the angle generator. At each stage $'s'$, the RAM address bits are rotated right by s -bits of butterfly counter $B(b_{s-1} b_{s-2} \ldots b_1 b_0 b_{n-2} b_{n-3} \ldots b_s)$and the control logic of the latch is driven by the sequence of the pattern; $b_{n-2} b_{n-3} \ldots b_s 0 \ldots 0$ (s "0" s).

2)X/Y Path of the proposed Architecture:-

The basic structure of proposed design for radix-4 CORDIC processor is shown in Fig.4.The proposed architecture requires $\left(\frac{n}{2}\right) = 8$ microrotation stages to implement x/y coordinates. The final $\frac{x}{y}$ coordinates (see Fig.4) are scaled using the scale factor computed in parallel with the micro rotations, which requires

Address Generation table of the proposed design for radix 2 – 8 point FFT

Table-1

Butterfly counter B b1b0	Stage 0		Stage 1		Stage 2	
	RAM address bits b1b0	Twiddle factor angles	RAM address bits b0b1	Twiddle factor angles	RAM address bits b1b0	Twiddle factor angles
00	00	0	01	0	00	0
01	01	π /4	10	0	01	0
10	10	2 π /4	01	2 π /4	10	0
11	11	3 π /4	11	2 π /4	11	0

adder/subtractor, $2:1$-multiplexer and an AND gate (Fig.4 b). Then the RBC converts the signed digit representation of the Radix-4 into binary form. Each micro rotation stage is designed by realizing the iteration equations,

$$X_{i+1} = X_i - \sigma_i * Y_i * 4^{-i}$$

$$Y_{i+1} = Y_i + \sigma_i * X_i * 4^{-i}$$

The Sign Digit adder/subtractor is controlled by the sign of σ_i value (see Fig. 4b). This σ_i values are realized by the prediction block (see fig.3).

X/Y Hardware realization:

The x/y block of the proposed system for each stage is composed by combination of a pair of signed binary adders, pair of registers at input and output and a pair of right-shift registers(see fig.). For the generated angle θ, the input vector $v_i = (X_i, Y_i)$is rotated to an output $v_{i+1} = (X_{i+1}, Y_{i+1})$with σ_ia predetermined direction vector, which assumes the direction of rotation.

3) σ Prediction block:

The σ -prediction block requires pipelined implementation of to compute the direction of microrotations for the given (generated by the Angle generator) target angle. Realizing, a linear relation between the rotation angle and the binary representation of directions of micro rotations [ ] given by

$$d = 0.5 * \theta + 0.5 * c_1 + \delta \qquad (4)$$

where;

$$c_1 = 2 - \sum_{i=0}^{\infty} (2^{-i} - \tan^{-1}(2^{-i})),$$

$$\delta = \sum_{i=0}^{n/3} (d_i * \epsilon_i) \text{ and } \epsilon_i = 2^{-i} - \tan^{-1}(2^{-i})$$

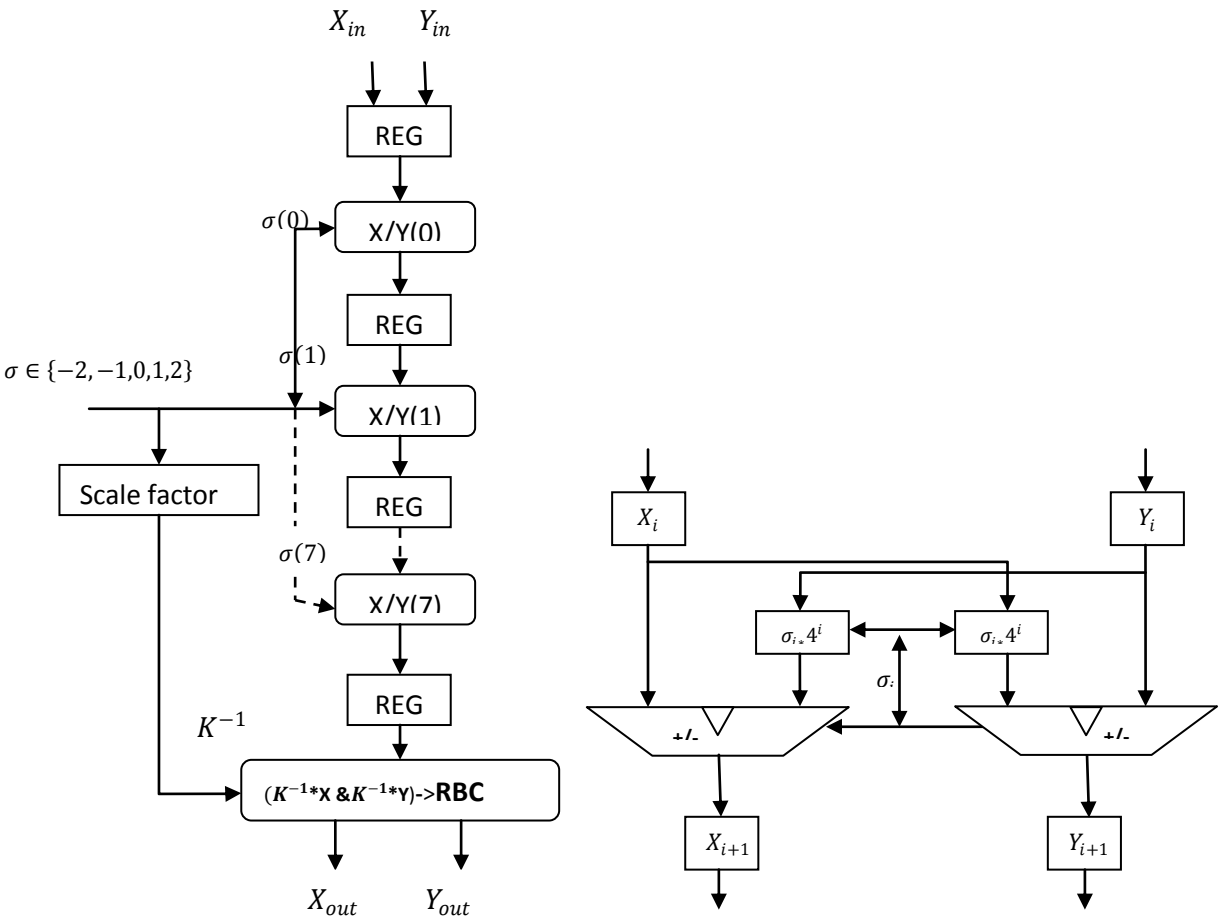

Fig 4. a) X/Y path ot the system, b) realization of X/Y path

is utilised to perform the direction of the vector. The direction of micro rotations in binary form is represented by the variable d' $(d-\theta)$ and θ has a linear relationship with negative slope being observed except for some θ value, called breakpoints [1, 27]. There by computing'd' from θ using simple addition operation as shown in the Fig. . The details of the various modules in the σ-precomputation block are as follows. For 16-bit precision, a(32 * 48) ROM is selected to store these offset values [ ]. The offset value δ is pre-computed for any input angle in the range$(-\pi/2, \pi/2)$ and the angle representing each of theseδ is referred to as θ_{ref}, which coincides to the summation of the first n/3 micro rotations. Therefore, the target angle has to be greater thanθ_{ref}. Each location of ROM holds a reference angle and two offset values(δ_k, δ_{k-1}). $\delta_k$corresponds to the reference angle and δ_{k-1}correspondsto the previous reference angle. For the given input angle,the offset value is obtained from the ROM using 2:1 multiplexer. Output of this multiplexer is controlled by the comparator, which generates sign signal '0' if $(\theta > \theta_{ref}$ for selecting $\delta_k)$ else '1' if$(\theta < \theta_{ref}$ for selecting $\delta_{k-1})$. The radix-4 arithmetic representation of $0.5c_1$ is added to 2's complement representation of 0.5θ and δ_{rom} using two adders to obtain$\sigma \epsilon \{-2,-1,0,1,2\}$. The delay of each adder is 2tFA, where tFA is one full adder delay. These $\sigma_i$values are stored in a buffer to transfer them to the corresponding stages of x/y path. Thus, the direction of rotations is predicted before the x/y path initiates the computation of x/y coordinates and following into elimination of the z path.

4) Scale factor

The scale factor of the radix-2 CORDIC rotator is a constant but in case of radix-4 CORDIC rotator it isn't, this is observable from the equation below

$$K^{-1} = \prod_{i=0}^{n/4} \left(\frac{1}{\sqrt{1+{\sigma_i}^2 * 4^{-2i}}} \right) \qquad (5)$$

where, $\sigma_i \epsilon \{-2,-1,0,1,2\}$. Here, we compute K^{-1} only for first $\left(\frac{n}{4}+1\right)=5$ iteration, as $k_i{}^{-1}$ $=\frac{1}{\sqrt{(1+4^{-2i})}}$becomes unity thereafter.For the present 16-bit implementation,it is adequate to approximate scale factor for the first 5- iterations only. Scale factor approximation is accomplished in parallel with the micro

rotations by bottling all possible scale factors in memory [1, 24]. From the expression above the $\sigma_i{}^2$ will have three values {0, 1, 4}.And so $\sigma_i{}^2$ can choose any one of the three values {0, 1, 4} in each iterationi, thereforefor 16-bit precision actually it is required to store 35 possible scale factors in ROM. But, we can cut down the size of the ROM to (27*16)using Taylor series expansion of the scale factor. Fori $\geq \left\lfloor \frac{n}{8} + 1 \right\rfloor$ with 16-bit precision, the scale factors corresponding to the first 3-iterations are obtained from(27*16) ROM. The 27 possible values for the scale factor $K_2{}^{-1}$ corresponding to the first three iterations are computed as

$$k_i{}^{-1} = \frac{1}{\sqrt{(1+\sigma_i{}^2 4^{-2i})}} \qquad (6)$$

The scale factor of the radix-2 CORDIC rotator is a constant but in case of radix-4 CORDIC rotator it isn't, this is observable from the equation below

$$K^{-1} = \prod_{i=0}^{n/4} \left(\frac{1}{\sqrt{1+\sigma_i{}^2 * 4^{-2i}}} \right) \qquad (7)$$

where, $\sigma_i \epsilon \{-2, -1, 0, 1, 2\}$. Here, we compute K^{-1} only for first $\left(\frac{n}{4} + 1\right) = 5$ iteration, as $k_i{}^{-1}$ $= \frac{1}{\sqrt{(1+4^{-2i})}}$becomes unity thereafter.For the present 16-bit implementation,it is adequate to approximate scale factor for the first 5- iterations only. Scale factor approximation is accomplished in parallel with the micro rotations by bottling all possible scale factors in memory [1, 24]. From the expression above the $\sigma_i{}^2$ will have three values {0, 1, 4}.And so $\sigma_i{}^2$ can choose any one of the three values {0, 1, 4} in each iterationi, thereforefor 16-bit precisionactually it is required to store 35 possible scale factors in ROM. But, we can cut down the size of the ROM to (27*16) using Taylor series expansion of the scale factor. For i $\geq \left\lfloor \frac{n}{8} + 1 \right\rfloor$ with 16-bit precision, the scale factors corresponding to the first 3-iterations are obtained from (27*16) ROM. The 27 possible values for the scale factor $K_2{}^{-1}$ corresponding to the first three iterations are computed as

$$k_i{}^{-1} = \frac{1}{\sqrt{(1+\sigma_i{}^2 4^{-2i})}} \qquad (8)$$

$$K_2{}^{-1} = \prod_{i=0}^{2} k_i{}^{-1} \qquad (9)$$

where, $k_i{}^{-1}$ is the scale factor for the i^{th} iteration.Using these values, the scale factors for the remaining two iterations are evaluated using a combinational logic as shown in Fig.7. The scale factors for the$k_3{}^{-1}$and$k_4{}^{-1}$(remaining two iterations) are derived by executing the shift and subtraction operations over the scale factor retrieved from ROM. This is carried out by realizing the first two terms of their Taylor series expansions. And then, the scale factor is coded using the canonical signed digit representation to carry out the final multiplication over the $\frac{x}{y}$ coordinates.

IV. Results & Discussion:

The algorithmic verification of the CORDIC algorithm is done with mat lab tool .(version 7.12 R2011a). The new z- path eliminated memory less CORDIC arc hitecture has been implemented on the available xc4Vfx12-12 sf363 of the Xilinx –FPGA (ISE13.4). The proposed architecture is done in Verilog. The number clock cycles required for computation of this new architecture is n/2 clock cycles with accuracy of 32 output bit precision. The latency of the system is n/2 clock cycles. The throughput rate is one valid result per eight clock cycles. The total power consumption of the new proposed implemented architecture is measured with X power software (ISE 13.4). The implemented design operates at 450.654 MHZ of clock rate with a power consumption of 175.90mW. The resource utilization of FPGA implementation of new memory reduced Z- path eliminated Radix-4 CORDIC processor is shown below and it's performance comparison is shown in Table[2]

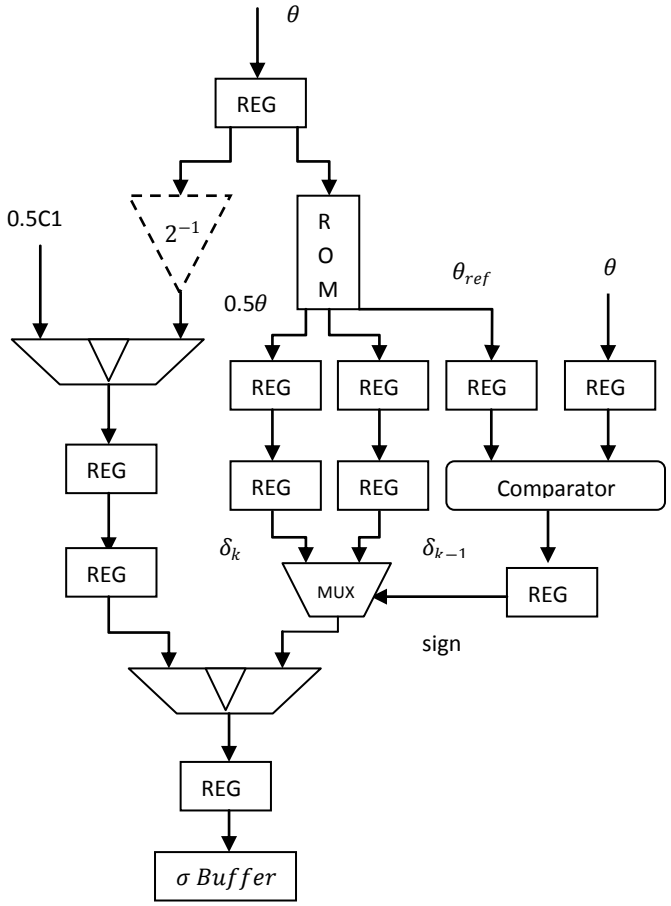

Fig.5 Realization of direction of rotation for an angle using σ −prediction

Selected Device : xc4vfx12 sf363-12

Number of Slices:	520	out of 5472 9%
Number of Slice Flip Flops:	936	out of 10944 8%
Number of 4 input LUTs:	899	out of 10944 8%
Number used as logic:	494	
Number used as Shift registers:	405	
Number of IOs:	65	
Number of bonded IOBs:	63	out of 240 26%
Number of GCLKs:	1	out of 32 3%

Performance comparison with [17] &[18]

Parameter	This work	[17]	[18]
Target Device	Xc4vfx12-sf363	XCV1000-6 BG560.	xc3s1500–4fg676
Slices	520	797	984
LUT's	899	1554	1907
Power	175.90mW	380mW	Not cited

VI. Conclusion:

A new memory less Z – path eliminated Radix-4 CORDIC architecture is presented in this paper. X and Y path pipelined and parallelism computed work was done , also eliminates the need for storing angles to avoid bottling the twiddle factor operation in FFT.A 16 bit new radix-4 memory less Z-path eliminated CORDIC architecture is done on FPGA platform(virtex 4).The latency of the proposed architecture is n/2 clock cycles with through put rate of one valid result per eight clock cycles. The entire new proposed architecture operates at the clock rate of 450.654 MHZ with power consumption of 175.90mW.The designed speed – power optimized 16 bit memory less Z-path eliminated Radix-4 can be applicable for FFT operation.

References:

[1] Sathish Sharma , Implementation of ParaCORDIC Algorithm and it's Applications in Satellite Communication ,2009 . International conference in Advances in Recent in Technologies in communication & computing

[2] Meng Qian, Applications of CORDIC Algorithm to Neural Networks in VLSI Design ,IMACS 2006,Beijing, China

[3] Javier Valls, The use of CORDIC in Software Defined Radios: A Tutorial, IEEE Communications Magazine, Sep '2006

[4] Ayan Benerjee, Swapna Banerjee, 'FPGA realization of a CORDIC based FFT processor for biomedical signal processing, Microprocessors and Microsystems, Feb 2011.

[5] .Bu-chin wang Digital signal processing Techniques and Applications in Radar Image processing.

[6] Bum Sikkim , Low power pipelined FFT architecture for Synthetic Aperture Radar, IEEE 39th Midwest Symposium, Circuits & Systems 1996.

[7] Sadat A , FFT for high speed OFDM wireless multimedia system, IEEE Circuits & Systems 2001 , MWSAS 2001

[8] Volder, J. (1959). The CORDIC trigonometric computing technique., IEEE Transactions on Electronic Computers

[9] B. Lakshmi , A.S. Dhar VLSI architecture for low latency radix-4 CORDIC, Elseiver Computers and Electrical Engineering, July 2011

[10] francisco j. jaime,enhanced scaling-free cordic, ieee transactions on circuits and systems—i: regular papers, vol. 57, no. 7, july 2010

[11] Erdal Oruklu & Xin Xiao & Jafar Saniie, Reduced memory low power architecture for CORDIC based FFT processors. J Sign Process Syst (2012)

[12] Lin, C., & Wu, A. (2005). Mixed-scaling rotation CORDIC (MSRCORDIC) algorithm and architecture for high- performance vector rotational DSP applications. IEE Transactions on Circuits and Systems I, 52(2011).

[13] Jiang, R. M. (2007). An area-efficient FFT architecture for OFDM digital video broadcasting. IEEE Transactions on ConsumerElectronics, 53(4), 1322–1326.

[14] Garrido, M., & Grajal, J. (2007). Efficient memory-less CORDIC for FFT Computation. In IEEE International Conference on Acoustics, Speech and Signal Processing, 2, 113–116), Apr.

[15] Xiao, X., Oruklu, E., & Saniie, J. (2009). Fast memory addressing scheme for radix-4 FFT implementation. In IEEE InternationalConference on Electro/Information Technology, EIT 2009,

[16] Xiao, X., Oruklu, E., & Saniie, J. (2010) Reduced Memory Architecture for CORDIC- based FFT. In IEEE International Symposium on Circuits and Systems,

[17] Anindya Sundar Dhar, Swapna Banerjee, Architectural design and FPGA implementation of radix-4 CORDIC processor, Elseiver, Microprocessors and Microsystems 34 (2010)

[18] M.G. Buddika Sumanasena, A scale factor correction scheme for the CORDIC algorithm, IEEE Transactions on Computers 57 (8) (2008)

[19] Vachhani, L., Sridharan, K., Meher, P.K., "Efficient CORDIC Algorithms and Architectures for Low Area and High Throughput Implementation," Circuits and Systems II: Express Briefs, IEEE Transactions on Jan. 2009,Volume: 56 , Issue: 1 Page(s): 61- 65

[20] Kuhlmann M, Parhi KK. P-CORDIC: A precomputation based rotation CORDIC algorithm. EURASIP J Appl Signal Process 2002;9:936–43.