IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
e-ISSN: 2278-2834,p- ISSN: 2278-8735.
PP 01-11
www.iosrjournals.org

FPGA implementation of Angle Generator for CORDIC Based High pass FIR Filter Design

Nutan Das[1], Swarnaprabha Jena[2], Siba Kumar Panda[2]

[1] *M.Tech Scholar,VLSI Design, Centurion University of Technology & Management, BBSR, Odisha*
[2] *Assistant Professor, Department of ECE, Centurion University of Technology & Management, BBSR, Odisha*

Abstract: *The design of high speed VLSI architecture introduced a large number of algorithmes for real-time Digital Signal Processing(DSP) and in the VLSI Technology, the realizations of this DSP algorithmes have been directed in many of the computations in Signal Processing and wireless communication appilications such as trigonometic and complex functions. This paper proposed a hardware efficient CORDIC based parallel architechture for the calculation of cosine and sine functions and High Pass FIR (Finite Impulse Response)Filter design using Very High Speed Integrated Circuit Hardware Description Language (VHDL). The generation of cosine sine functions of CORDIC algorithm have synthesized, simulated and tested it on Xilinx FPGA SPARTAN 3E kit.The code have synthesized using Xilinx ISim 14.4 simulator software. This paper also described the angle to 15bit binary conversion, analysis of all output values and the power utilization summary .Finally, magnitude plot of HighPass FIR Filter is plotted and analysed.*

Keywords: *CORDIC, FPGA, FIR FILTER, SIGNAL PROCESSING, VLSI SIGNAL PROCESSING, VHDL*

I. Introduction

In advanced VLSI technology have stimulated a great interest in developing special purpose processor architecture arrays to facilitate real-time signal processing. The reductions in the hardware cost, less area, high speed and less power consumption are motivated the development of various architectures to design filters in Digital Signal Processing. Generally, DSP algorithms use the calculation of elementary functions which require high computational power.

Now, Field-Programmable Gate Array (FPGA) is introduced hardware technology for DSP systems offer the capability to develop the most suitable architecture for the computational, memory and power requirements of the DSP applications. For FPGA implementation the DSP system produced highly parallel, pipelined and hybrid architecture. These Filters are used to demonstrate the mapping, introduced retiming and the low power optimizations are demonstrated by using a FFT-based applications and development.

In the CORDIC(Coordinate Rotation Digital Computer), J.E. Volder [1], Rotation is a 2-D(dimensional) vector and a target angle is divided into desired rotation angle into the weighted sum of a set of predefined elementary rotation angles in which through each rotation can be realized with shift and add/sub operations. Also, the CORDIC algorithm is again extended [2] to proposed a unified algorithm for computation of rotation in and hyperbolic coordinate systems, circular, linear, embedding coordinate system. These functions are implemented for many applications like digital signal processing [3].

CORDIC used mainly simple shift and add operations[4], that performed several computational tasks are calculation of trigonometric functions(cosine,sine), hyperbolic functions(htan,hcos) and logarithmic functions, real and complex functions, and simple arithmetic functions are multiplications, division, square-root functions Now CORDIC Algorithms are used in the area of signal and image processing and communication systems, robotics and 3-D(Dimensional) graphics, Fast Fourier Transforms(FFT), Filters, Computer Graphics[5] and Robotics [6]. CORDIC offers high computational applications: computer graphics, in which a combination of scaling and rotations are required for real time world. For fast computation programming CORDIC is again used in the field of Robotics.

II. Background And Literature Review

Amritakar Mandal *etal.* [7] described the design of pipelined architecture for the computation of Sine and Cosine values and it is based on application of a specific CORDIC processor. The design of CORDIC is mainly based on the circular rotation mode gives a high system throughput by reducing latency in each individual pipelined stage.

Lakshmi.B *etal.* [8] explained a low latency field programmable gate array implementation of an unfolded architecture for the implementation of rotational CORDIC algorithm. Here the computational device was highly suitable for the implementation of customized hardware in portable devices and in which a large parallelism and low clock rate were utilized for low power consumption.

Siba Ku panda *etal.* [9, 10] explained an innovative idea of designing multiplier circuit for various VLSI signal processing applications.

Ray Andraka [11] has summarized that by using CORDIC Algorithm some functions like sine, cosine, tangent, circular, linear, hyperbolic functions, Cartesian to polar transformations and inverse functions was implemented. For FPGA Implementation next time this paper also described as bit serial, iterative and online CORDIC architecture design.

Velmurugan [12] explained a proposed low power analog and mixed mode implementation of particle filter for target tracking. Based on CORDIC Algorithm[13] the particle filter algorithm was used to exploits addition, multiplication, Gaussian and Arctan calculations.

X. Hu. *etal*. [14] has described a proposed unified CORDIC algorithm based on the parallel and pipelined CORDIC processor. J. Sudhaa *eta*l.[15] has proposed the potential application of the proposed 2D Gaussian function includes image enhancement, smoothing, edge detection, filtering etc and to gain real time performance, the architectures developed exploits high degrees of pipelining and parallel processing .

N. Takagi, *etal* [16] described a proposed Double rotation and correcting rotation methods to implement constant scale factor CORDIC in which result was 50% increasing in number of iterations. For that this increase in latency is reduced by proposed branching algorithm.

J. Duprat etal. [17] described additional CORDIC module to perform rotations in both directions, if the direction cannot be determined using intermediate results. The main demerits of branching method is the necessity of performing two conventional CORDIC iterations in parallel, which consumes more silicon area than the conventional methods. Also, this method gives a faster implementation than [16].

M. Chakraborty, *etal.*[18] explained a class of pipelined CORDIC architectures for the LMS-based transversal adaptive filter. Introducing an alternate formulation of the LMS algorithm, obtained by expressing the mean square error as a convex function of a set of angle variables and monotonically related to the filter tap weights. Also proposed architectures employed a micro-level pipelining and are adjustable to strike tradeoffs between throughput efficiency vis-a-vis hardware complexity.

Shanmuga Kumar M. *etal.* [19] summarized a proposed the high precision and hardware efficient CORDIC structure for handheld scientific calculator. Tsao Y C, *etal*.[20] presented the realization of area efficient architectures using Distributed Arithmetic (DA) for implementation of Finite Impulse Response (FIR) filter.Also described the performance of the bit-serial and bit parallel DA along with pipelining architecture and different quantized versions are analyzed for FIR filter Design.

Shrikant Patel ,*etal* [21] presented the implementation of Modified Distributed Arithmetic, and introduce it into the FIR filters design, and then presents a 31-order FIR low-pass filter using Modified Distributed Arithmetic and save considerable MAC blocks to decrease the circuit scale.

M.Chakrapani ,*etal* [22] explained the mode of operation of CORDIC algorithm and Control CORDIC Angle correction and Quadrant correction. Latency can be further reduced by another CORDIC algorithm techniques. By angle selection scheme the Overshoot problem in original CORDIC has been overcome and complexity of hardware also increases for the increasing of number of bits.

The contribution of this paper is based on a new and modified architecture in which it needs less area for implementation and less delay output. The need of hardware efficient algorithms in the advanced technologies For the fast signal processing requirements .So, the current trend is back toward hardware efficient algorithms, many of those applications require elementary calculations like sin and cosine function. So, in this paper i am presenting a angle generator using CORDIC algorithm and its performance reports. The parallel architecture takes the angle as input and gives both sine and cosine for the given input in predetermined number of micro rotations. These micro rotations are decided by the accuracy demanded by the application. Here our objective is design to plot the magnitude curve using the frequency response equation of a FIR Filter using CORDIC algorithm in VHDL.

Section I gives brief introduction, section II defines the Background and Literature Review. The Basic CORDIC algorithm and architecture are briefly studied in section III, The FPGA Implementation, Simulation and Synthesis results are discussed in section IV and finally conclusion is explained in section V.

III. Basic Cordic Algorithm And Architechture

Coordinate Rotation Digital Computer (CORDIC) ,firstly implemented in1959 by J.E.Volder [1],is an ancient principles of two-dimensional geometry. CORDIC is a typical iterative algorithm, is used to accelerate many digital functions and applications in signal processing. This algorithm may be realized as an iterative form of add, sub and shift operations. The two basic modes in which the different functions are computing and these are a the rotation mode and the vectoring mode. The algorithm can be realized as an iterative sequence of additions or subtractions and shift operations by using the two modes, which are rotated by a fixed rotation angle (micro rotations) .

In Rotation mode is shown in Fig-2 , we can rotate (1,0) by an angle φ degrees to get (x, y)

$$\begin{aligned} x &= x.\cos(\phi) \\ y &= y.\sin(\phi) \end{aligned} \qquad (1)$$

$$\begin{aligned} x' &= x.\cos(\phi) - y.\sin(\phi) \\ y' &= y.\cos(\phi) + x.\sin(\phi) \end{aligned} \qquad (2)$$

Rearranged as the equations are

$$\begin{aligned} x' &= \cos(\phi).[x - y.\tan(\phi)] \\ y' &= \cos(\phi).[y_i + x.\tan(\phi)] \end{aligned} \qquad (3)$$

Rewrite in terms of a_i: $(0 \le i \le n)$

$$\begin{aligned} x_{i+1} &= \cos(\phi_i).[x_i - y_i.d_i.\tan(\phi_i)] \\ y_{i+1} &= \cos(\phi_i).[y_i + x_i.d_i.\tan(\phi_i)] \end{aligned} \qquad (4)$$

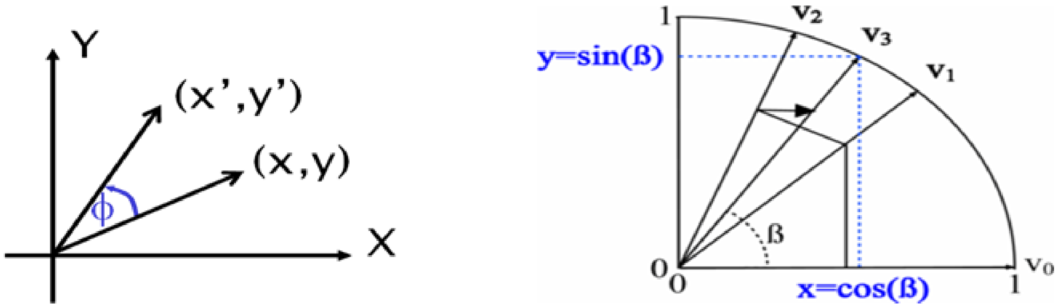


Figure- 1 (a) and (b) *Vector mode by the angle φ*

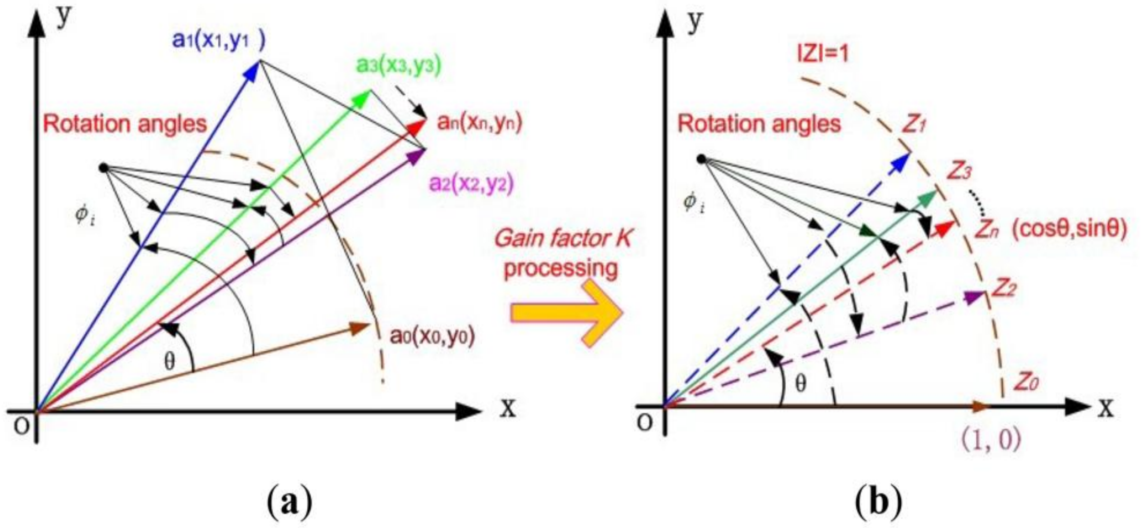


Figure- 2 (a) and (b) *Rotation mode by the angle Ɵ*

We can restrict the rotation angle $\tan(\alpha_i)$ to $\tan(\phi_i) = \pm 2^{-i}$, then the multiply operation by tangent part is reduced to shift operations. The new iterative rotation can be defined as

$$\begin{aligned} x_{i+1} &= K_i.[x_i - y_i.d_i.2^{-i}] \\ y_{i+1} &= K_i.[y_i + x_i.d_i.2^{-i}] \end{aligned} \qquad (5)$$

Where, i=0,1,2,3,4……n.

Removing the scaling factor K$_i$ from the iterative part yields a simple shift-add algorithm for rotation CORDIC. After several iterations, the product of K$_i$ factors will converge to a constant coefficient 0.607252.The exact gain depends on the number of iterations:

$$An = \Pi n\sqrt{1+2^{-2i}} \approx 1.646762 = \frac{1}{Ki} \qquad (6)$$

Then the angle accumulator is added to the Equation [7],below

$$\begin{aligned} x_{i+1} &= x_i - y_i.d_i.2^{-i} \\ y_{i+1} &= y_i + x_i.d_i.2^{-i} \qquad (7) \\ z_{i+1} &= z_i \_ d_i.\tan^{-1} 2^{-i} \end{aligned}$$

Where

$$d_i = \begin{cases} -1, z_i < 0 \\ +1, otherwise \end{cases}$$

The angle accumulator is initialized with the input rotation angle and the rotation direction at each iteration i is decided by the magnitude of the residual angle in the angle accumulator. If the residual angle is positive, then

$$d_i = \begin{cases} +1, z_i >= 0 \\ -1, otherwise \end{cases}$$

A CORDIC has three inputs x0,y0,z0 and depending on the inputs to the Rotation CORDIC and various results can be produced at the outputs are Xn, Yn, Zn shown in Fig:2

After several iterations it will produce the results according to a CORDIC Block is shown in Fig-3.

$$\begin{aligned} x_n &= A_n\left(x_0 - y_0 \tan z_0\right) \qquad (8) \\ z_n &= 0 \\ y_n &= A_n\left(y_0 + x_0 \tan z_0\right) \end{aligned}$$

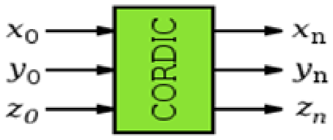


Figure-3 A CORDIC Block

For convergence of φ to 0 choose di=sign Zi

If we can start with X0=1/k and y=0, at the end of the process,we find Xn=Cos Z0 and Yn= SinZ0 and the domain of convergence is lies between -99.7<Z0<99.7.

A. Using CORDIC in Vectoring mode is shown in Fig-2, we can write the expression as below:

$$\begin{aligned} x_{i+1} &= x_i - y_i.d_i.2^{-i} \qquad (9) \\ y_{i+1} &= y_i + x_i.d_i.2^{-i} \\ z_{i+1} &= z_i \_ d_i.\tan^{-1} 2^{-i} \end{aligned}$$

And we can solve as

$$\begin{aligned} x_n &= An\sqrt{x^2 + y^2} \qquad (10) \\ y_n &= 0 \\ z_n &= z_0 + \tan^{-1}(\frac{y_0}{x_0}) \end{aligned}$$

For convergence of Yn to 0, choose

$$\begin{aligned} d_i &= -\text{sgn}(x_i y_i) \\ z_n &= \tan^{-1} y_0 \end{aligned}$$

If we start with x0=1 and z0=0, we find

$$d_i = \begin{cases} +1, y_i < 0 \\ -1, otherwise \end{cases}$$

3.1 Required Angle

At first pre -computation of $\tan\alpha_i = (2^{-1})$ **or** $\alpha_i = \tan^{-1}(2^{-1})$ value then we obtained binary value of each required angle. Here we use 10 iterations, so each iteration we use the α_i value from the TABLE-1 as shown below.

Table-1 Required Angle For Each Iteration

ITERATION(i)	REQUIRED ANGLE $\alpha_i = \tan^{-1}(2^{-1})$	$\tan\alpha_i$	OBTAINED BINARY VALUE
0	45	0.7854	001000000000000
1	26.565	0.4636	000100101110010
2	14.036	0.2450	000010011111101
3	7.125	0.1244	000001010001000
4	3.5763	0.0624	000000101000101
5	1.7899	0.0312	000000010100010
6	0.8951	0.0156	000000001010001
7	0.4476	0.0078	000000000101000
8	0.2238	0.0039	000000000010100
9	0.1119	0.0019	000000000001010

3.2 Working Cordic Architechture

The working CORDIC parallel Architecture figure is given below. This architecture consists of Add block, add/sub block, comparator and shift blocks as shown in Fig-4.We have assumed that, if the input angular value<=45 degree, then inputs (x, y) are (1,0) or else (0,1).Then compare this angle with a reference angle (30 degree) and right bit-shifted operation is occurred depending upon the iteration step(i), x(i) and y(i) and then added or subtracted depending upon the value of comparator circuit(z0) with x(i) and y(i) respectively to generate x(i+1) and y(i+1).Again we repeat the 10 iterations to getting the desire output values.

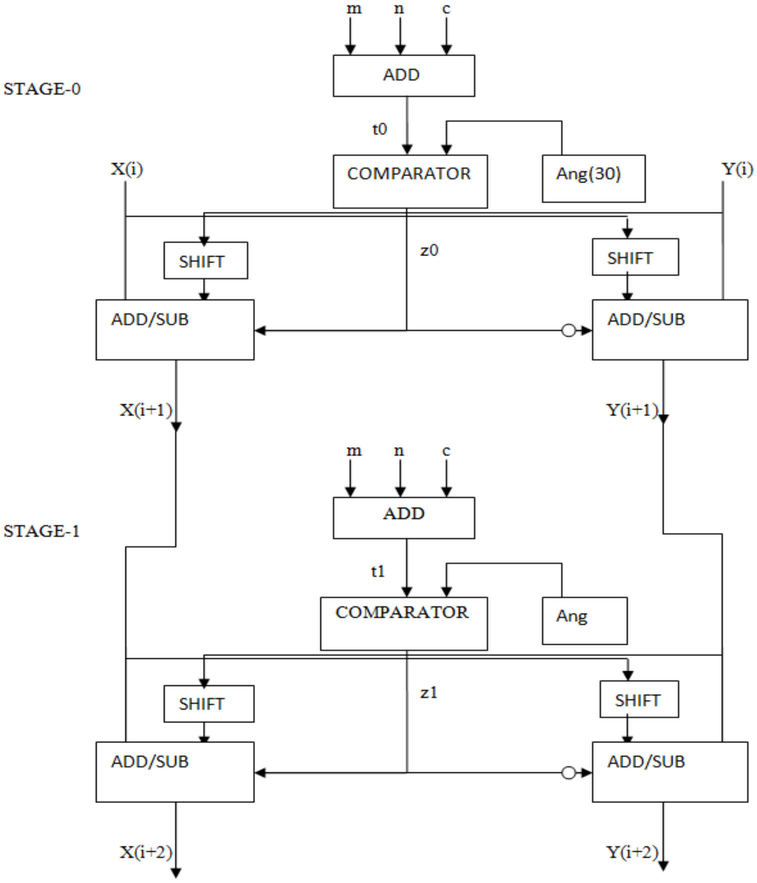


Figure-4 Working Parallel CORDIC Architecture

3.3 Cordic Based High Pass FIR Filter

The Block diagram of CORDIC based high pass FIR filter is shown in FIG-5.Here the main blocks are Bit-shift, Add/Sub and CORDIC and the given inputs to the HP FIR filter design is in a 15-bit binary format .The angular inputs w,2w,3w are provided through the CORDIC blocks and generate the Cosine angle and finally using the Add/Sub blocks produce the desire output . Here we have chosen an arbitrary frequency response equation of an High Pass FIR filter is given in equation (11).

$$\overline{H}(e^{jw}) = 0.75 - 0.45\cos w - 0.31\cos 2w - 0.15\cos 3w \qquad (11)$$

CORDIC based High Pass FIR Filter

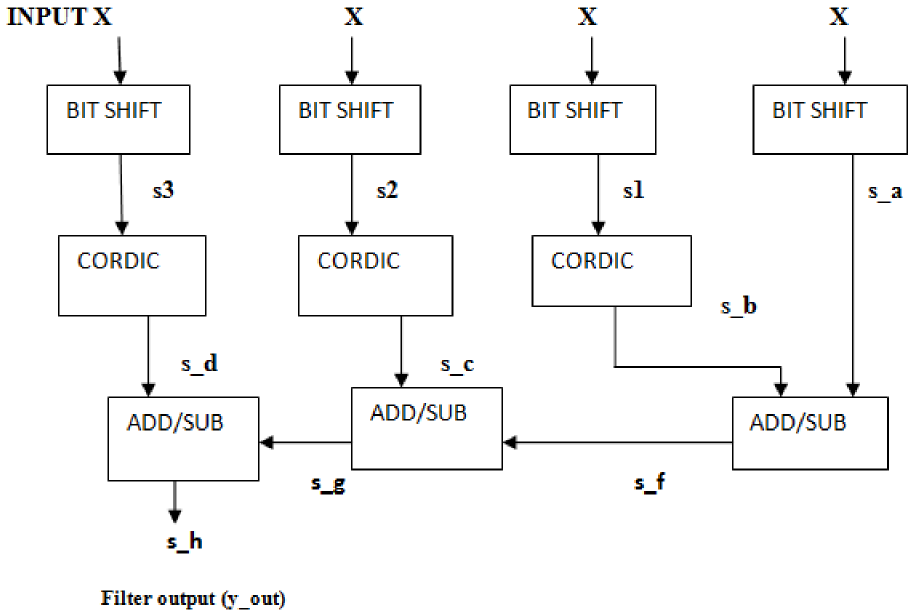


Figure-5 Block diagram of High pass FIR Filter

IV. FPGA Implementation And Simulation Result

In this paper, the FPGA implementation is done for generating the sine and cosine values in binary format of given angle (30 degree) and taking the input (1,0)of X and Y value in 2bit form using CORDIC algorithm as shown in Fig-6 and RTL schematic is also shown in Fig-7.Both 15 bit CORDIC architecture and CORDIC based FIR Filter module have implemented by using Xilinx ISE Design 14.4 as shown in Fig-8 and Fig-9, described in VHDL language and verified and synthesized the functionality using Xilinx ISim simulator as shown in Fig-10 and Fig-11. The top-level RTL schematic figures are shown below.

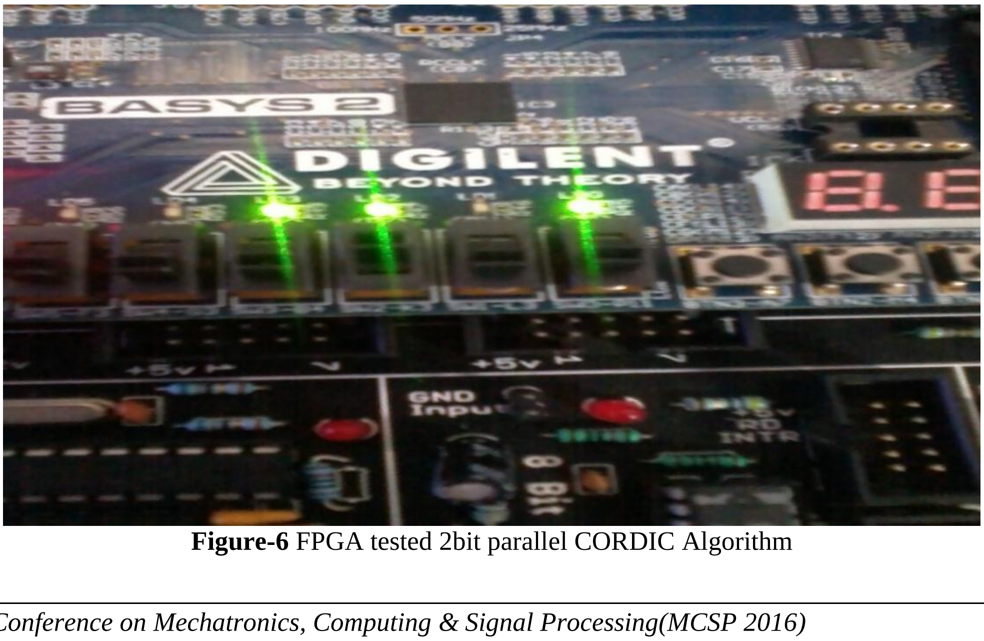


Figure-6 FPGA tested 2bit parallel CORDIC Algorithm

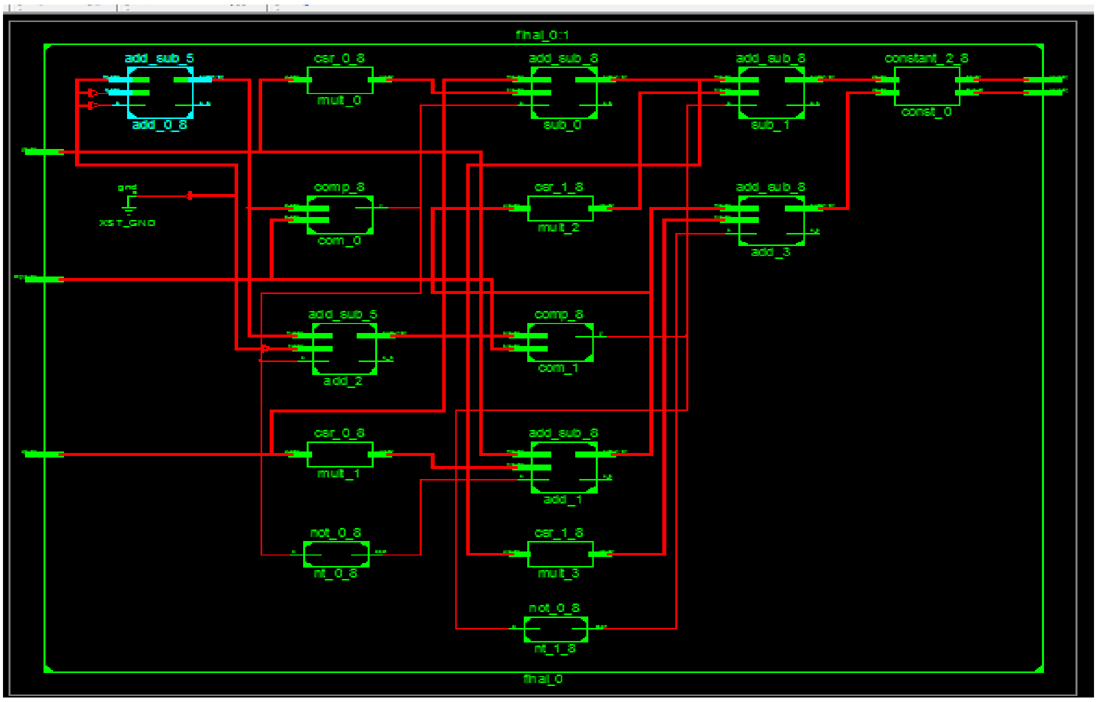


Figure-7 RTL Schematic of Parallel CORDIC Top level 2bit Architecture

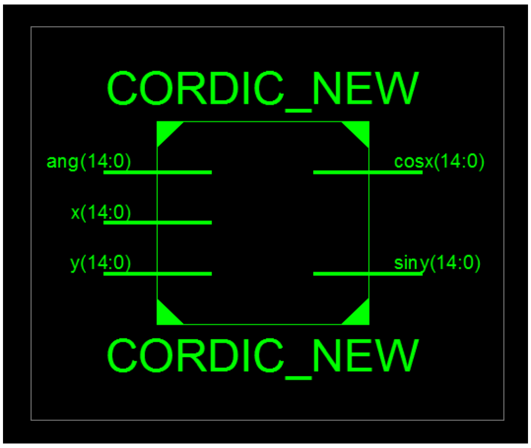


Figure-8 Top- level RTL CORDIC Architecture

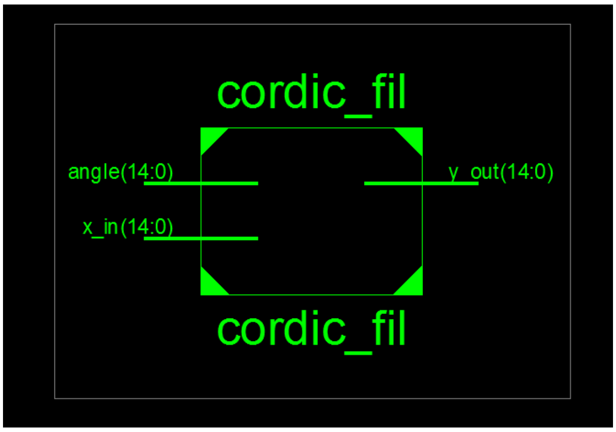


Figure-9 Top- level RTL CORDIC FIR Filter

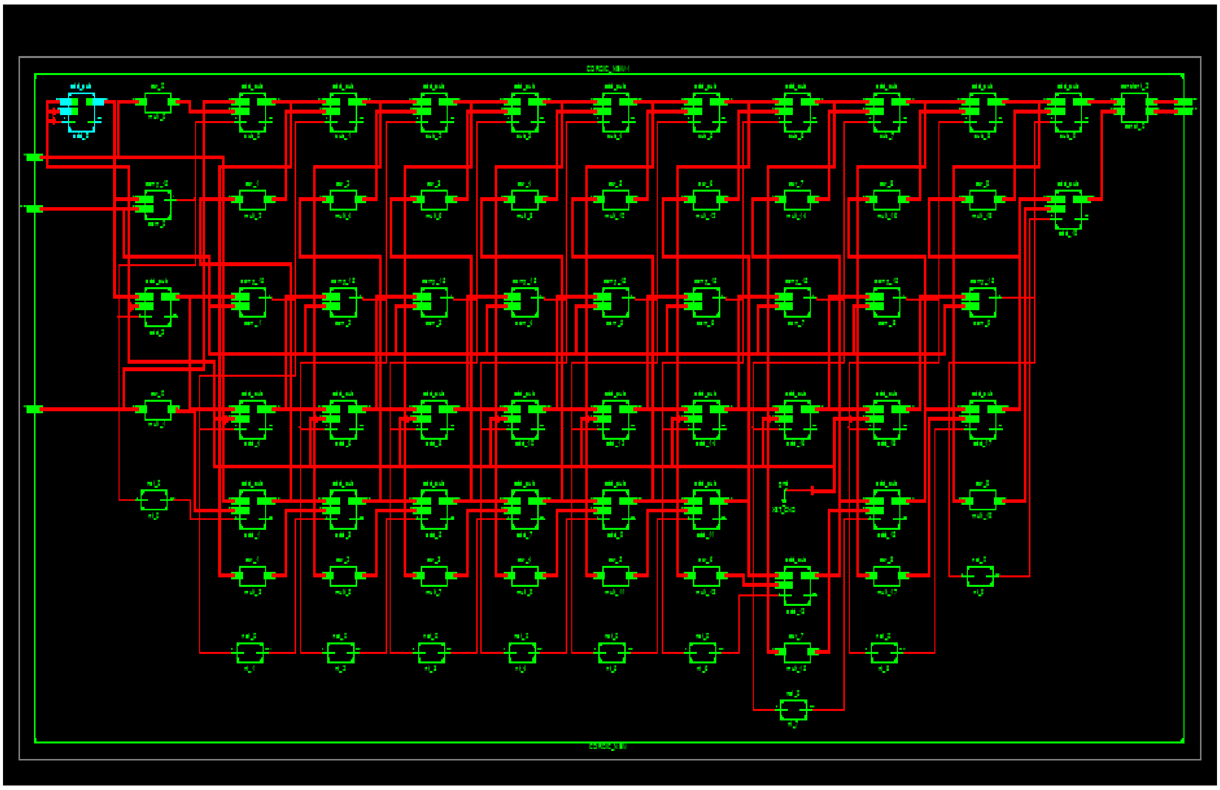


Figure-10 RTL Schematic of CORDIC Architecture

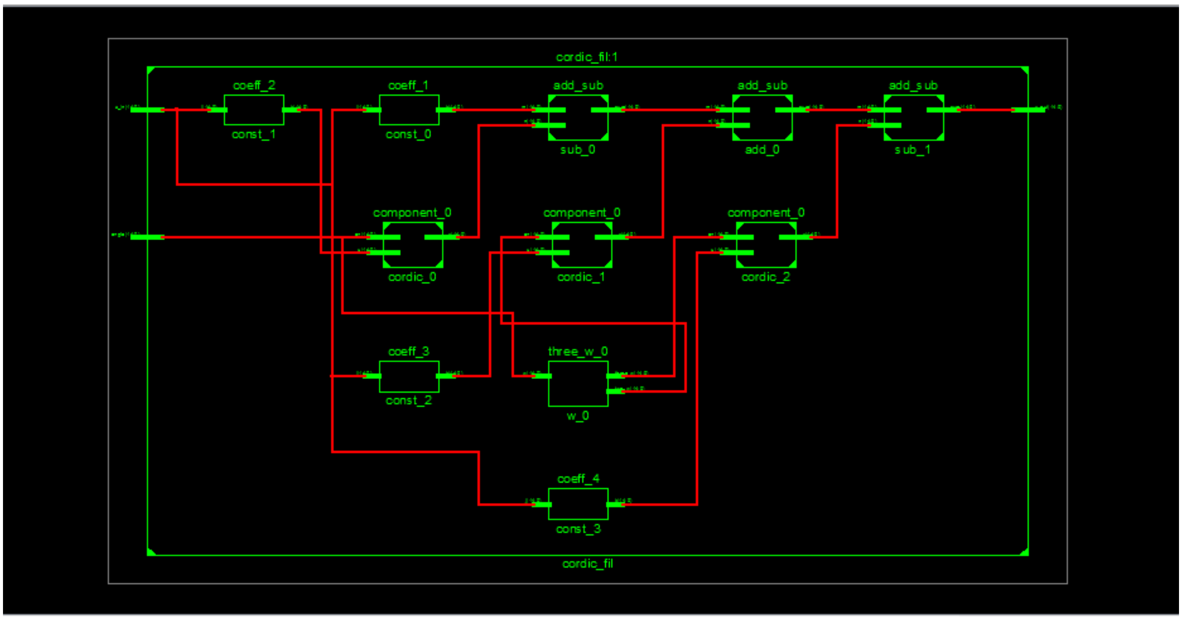
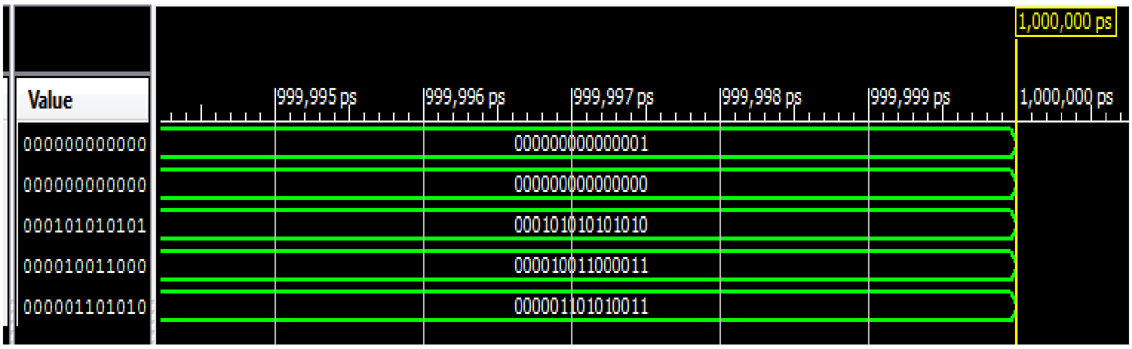


Figure-11 RTL Schematic of FIR Filter

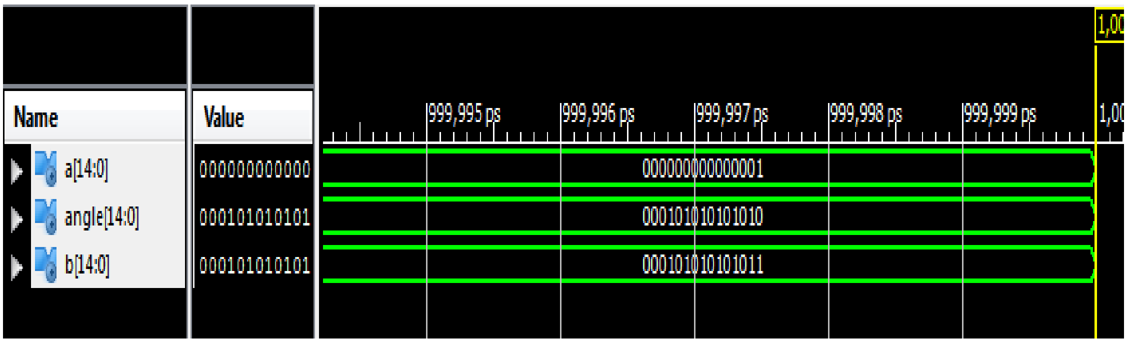
4.1 Output Simulation Result

The output simulation result for parallel CORDIC Architecture (CORDIC_NEW) is shown in Fig-12 and also output simulation result for FIR Filter(30 degree) and (25 degree) are shown in Fig-13 and Fig-14.



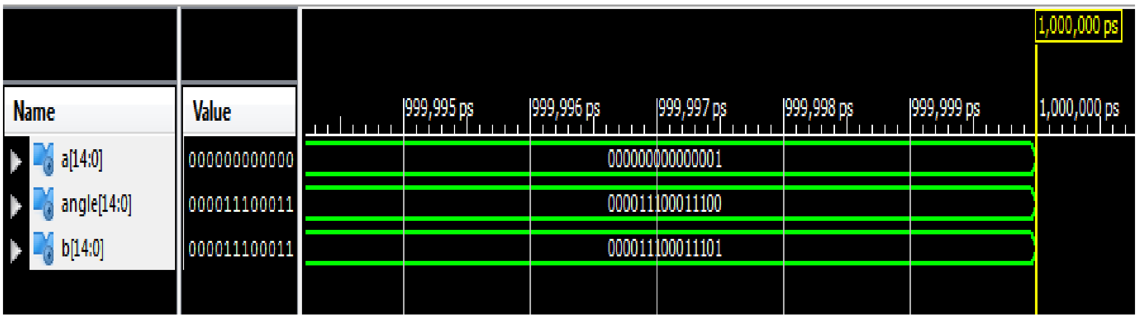
```
Value
000000000000    000000000000001
000000000000    000000000000000
000101010101    000101010101010
000010011000    000010011000011
000001101010    000001101010011
```

Figure-12 Simulation Result For Parallel CORDIC Architecture (CORDIC_NEW)



```
Name          Value
a[14:0]       000000000000    000000000000001
angle[14:0]   000101010101    000101010101010
b[14:0]       000101010101    000101010101011
```

Figure-13 Simulation Result For(30 Degree) FIR Filter



```
Name          Value
a[14:0]       000000000000    000000000000001
angle[14:0]   000011100011    000011100011100
b[14:0]       000011100011    000011100011101
```

Figure14 Simulation Result For(25 Degree) FIR Filter

4.2 Magnitude plot of High Pass FIR Filter

The Magnitude plot of high pass filter is shown in Fig-15, described the accuracy between the educational value and output calculated values of generated angles (10 degree to85 degree) from CORDIC based HP FIR Filter Design circuit using VHDL language. The plot clearly reflects the accuracy of the output which is generated from the CORDIC based FIR Filter frequency response equation (11).

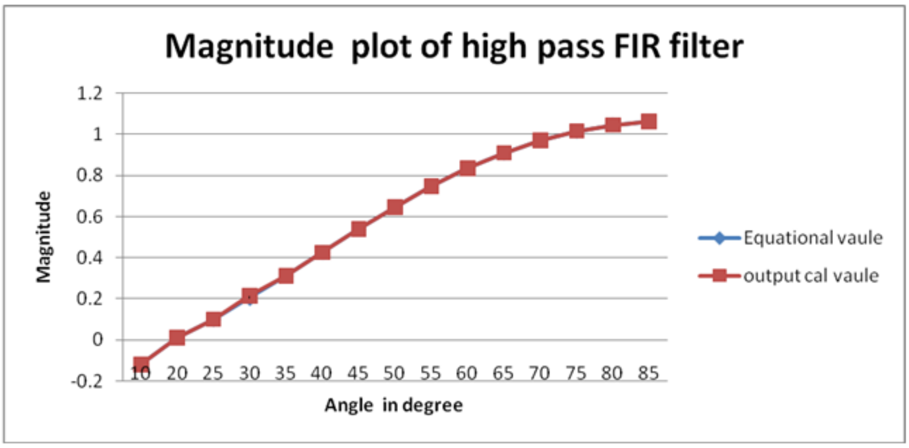
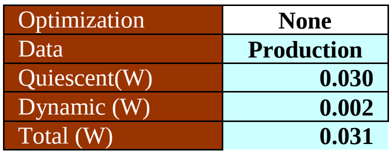


Figure-15 Magnitude Plot Of High Pass FIR Filter

4.3 Power Summry

The Power Summery of Parallel CORDIC Algorithm is given in Fig-16.The total Power utilization is 0.031W.



Optimization	None
Data	Production
Quiescent(W)	0.030
Dynamic (W)	0.002
Total (W)	0.031

Figure-16 Power Summery of Parallel CORDIC Algorithm

V. Conclusion

We have successfully simulated and imlpemented the CORDIC algorithm in FPGA for calculating the Cosine and Sine of an angle in 15 bit binary format on Xilinx Spartan3E (XC3S100Ecp132-5) device using ISE Design Suite 14.4,ISim Simulator of VHDL language.Also we have plotted the magnitude curve of a High Pass FIR Filter and this error less curve was generated for output accuracy between equational calculation output values and generated output of CORDIC_FIL values .In future CORDIC algorithm can be further extended for clock- pipellined architecture in VERILOG language and to calculate more complex and higher order problems and digital filter design in DSP and VLSI domain with more acuracy ,high speed and low cost.The number of Input Output Blocks utilization in both cases(CORDIC_NEW ,CORDIC_FIR) are 90% and 68% respectfully and the total power utilization is 0.030.

References

[1]. Volder J.E., “The CORDIC Trigonometric Computing Technique” ,*IRE Transactions Electronic Computing,* Volume-EC-8,PP.-330-334,1959.

[2]. J.S. Walther,“*A unified algorithm for elementary functions”,* in AFIPS Spring Joint Computer Conference, May 1971, PP.379-385.

[3]. Y.H. Hu., “*CORDIC-based VLSI architectures for digital signal processing”*, IEEE Signal Process.Mag.9(3) 1992,PP.16-35.

[4]. P.K. Meher, J. Valls, T.B.Juan, K. Sridharan and K.Maharatna, “*50 Years of CORDIC: Algorithms, Architectures, and Appilications*” IEEE Transactions on circuits and systems irregular papers, vol. 56, no.10, PP.9, Sept.2009.

[5]. F. Koscsis and J. Bohme, “*Fast algorithms and parallel structures for form factor evaluation”,* The Visual Computer, no.8,PP.205-216,1992.

[6]. M. Kamey ama, T. Amada, and T. Higuc hi, “*Highly parallel collision detection processor for intelligent robots*", IEEE Journal of Solid-State Circuits, vol. 27, no. 4, pp. 500-506, 1992.

[7]. Amritakar Mandal, K. C. Tyagiand Brajesh Kumar Kaushik, "VLSI Architecture Design and Implementation for Application Specific CORDIC Processor," *ARTCOM '10 Proceedings of the 2010 International Conference on Advances in Recent Technologies in Communication and Computing*, PP. 191-193.

[8]. Lakshmi.B and Dhar. A.S, "*FPGA implementation of a high speed VLSI architecture for CORDIC,*" *TENCON 2009,IEEE Region 10 Conference on* 23-26 Jan. 2009, PP: 1- 5.

[9]. Arati Sahu,Siba Ku.Panda,Swarnaprabha Jena “HDL Implementation and Performance Comparison of an Optimized High Speed Multiplier”, IOSR Journal of VLSI and Signal Processing (IOSR-JVSP),Volume 5, Issue 2, Ver. I (Mar - April. 2015), PP 10-19.

[10]. Siba Ku.Panda,R.Das,S.k Saifur Raheman and T.R Sahoo, “VLSI implementation of Vedic Multiplier using Urdhva-Tiryakbhyam sutra in VHDL environment:A Novelty”, IOSR Journal of VLSI and Signal Processing (IOSR-JVSP),Volume 5, Issue 1, Ver. III (Jan - Feb. 2015), PP 17-24.

[11]. Andraka R. A, “A *Survey of CORDIC A lgorithms for FPGA based C omputers*,” *Proceedings of the 1998A C M /SIGDA sixth International Symposium on FPGAs*, , Monterey, California, 22-24 Feb., 1998, pp. 191-200.

[12]. Velmurugan R, Cevher V, and McClellan J. H, “*Implementation of batch-based Particle Filters for Multisensory Tracking*”, ,IEEE *Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, Dec., 2007, U. S. Virgin Islands.

[13]. Bhattacharyya K, Biswas R, Dhar A. S, “*Architectural Design and FPGA Implementation of Radix-4 CORDIC Processor*”,*Microprocessors and Microsystems (MICPRO, Elsevier) Embedded Hardware Design*, 2010,Vol. 34, PP. 96-10.

[14]. X. Hu, R. Huber, S. Bass, “*Expanding the Range of Convergence of the CORDIC Algorithm*”, *IEEE Transactions on Computers*, Jan., 1991, Vol. 40, No. 1, PP. 13-21.

[15]. Sudhaa, M. C Hanumantharajub, V. Venkateswarulua, Jayalaxmi Hc, “*A Novel Method for Computing Exponential Function Using CORDIC Algorithm*”, International Conference on Communication Technology and System Design 2011,PP-519 – 528.

[16]. N. Takagi, T. Asada, and S. Yajima, “*Redundant CORDIC methods with a constant scale factor for sine and cosine computation*”, IEEE Trans. Comput., vol. 40,no. 9, PP. 989-995,Sept., 1991.

[17]. J. Duprat, and J. M. Muller, “*The CORDIC algorithm: new results for fast VLSI implementation*”, IEEE Trans.Comput.vol. 42, no. 2, PP. 168-178, Feb. 1993.

[18]. M. Chakraborty, A.S. Dhar, S. Pervin, "*CORDIC realization of the transversal adaptive filter using a trigonometric LMS algorithm*", *ICASSP*, 2001, Acoustics, Speech, and Signal Processing, IEEE International Conference on, Acoustics, Speech, and Signal Processing, IEEE International Conference on 2001, PP. 1225-1228,

[19]. Shanmuga Kumar M a, Mohamed Asan Basiri M b, and Noor Mahammad Sk ,"*High Precision and High Speed handheld Scientific Calculator Design using hardware based CORDIC Algorithm*," Procedia Engineering 64 (2013) PP. 56 – 64.

[20]. Tsao Y C and Choi K. "*Area-Efficient Parallel FIR Digital Filter Structures for Symmetric Convolutions Based on Fast FIR Algorithm",*IEEE Transactions on Very Large Scale Integration (VLSI) Systems,, PP(99),2010.

[21]. Shrikant Patel,"*Design and implementation of 31-order fir low-pass filter using modified distributed arithmetic based on FPGA*", International Journal of Advanced Research in Electrical,Electronics and Instrumentation Engineering, Vol. 2, Issue 10, October 2013.PP:5060-5072.

[22]. M.Chakrapani, J.S.S.Ramaraju, Ch.N.L.Sujatha, "*Implementation of Cordic Algorithm for FPGA Based Computers Using Verilog*", International Journal of Advanced Research in Electrical Electronics and Instrumentation Engineering, Vol. 3, Issue 8, August2014,PP:11487-11496.