IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
Volume 4, Issue 2, Ver. II (Mar-Apr. 2014), PP 29-37
e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197
www.iosrjournals.org

A Modified CORDIC Processor for Specific Angle Rotation based Applications

N.V.Shimna[1], E.Konguvel[2], Dr.J.Raja[3]
[1,2,3]*Department of Electronics and Communication Engineering,*
[1,2]*Dhanalakshmi Srinivasan College of Engineering, Coimbatore, India.*
[3]*Adhiparasakthi Engineering College, Chennai, India.*

Abstract: *CORDIC algorithm provides an efficient way for vector rotation in a plane through a fixed and known angle with high level of accuracy. CORDIC requires only simple shift add operation to estimate the basic elementary functions like trigonometric operations, multiplication, division and some other operations like logarithmic functions, square roots and exponential functions. This rotation of a given vector (x_i, y_i) is examined by means of a sequence of rotations with fixed angles which results in overall rotation through a given angle or result in a final angular argument of zero. A hardwired pre-shifting scheme in barrel shifters is introduced here to reduce the area and time complexities. An iterative form of calculation is done here for Co-ordinate and angle measurement of the vectors. In this paper simple shifters and adders / subtractors are used for the calculations. A look up table is used to set the values of the constants according to the demand of angle setting for the algorithm. To reduce the complexity and number of resources used single rotation of vectors is adapted. CORDIC is a good choice for hardware solutions such as FPGA in which cost minimization is more important than throughput maximization. Simulation is done using ModelSim software.*
Keywords: *CORDIC, ANGLE ROTATION.*

I. Introduction

CORDIC stands for CO-ordinate Rotation DIgital Computer. It was introduced in 1959 by J.E.Volder for implementing a real-time navigation computer for aeronautical applications. The algorithm was initially formulated for computing the values of trigonometric functions. More recently efficient hardware technique known as BKM for computing complex exponentials and trigonometric functions was proposed and has since been very widely applied.

In the present time CORDIC algorithm have a number of applications in the field of robotics, communication, 3-D graphics, games, animation, signal processing and a lot more. The main concept of this algorithm is based on the very simple and long lasting fundamentals of two-dimensional geometry. This algorithm needs only adders, shifters and comparators for computing a wide range of elementary functions.

There are two modes of operation for a CORDIC rotator. The first mode is called rotation mode which rotates the input vector by a specified angle (given as an argument).The second mode is called vectoring mode which rotates the input vector to the *x* axis while recording the angle required for making that rotation. The angle accumulator is initialized with the desired rotation angle in rotation mode. In the angle accumulator the rotation decision at each iteration is made to diminish the magnitude of the residual angle.

In the vectoring mode, the CORDIC rotator rotates the input vector through whatever angle is necessary to align the result vector with the x axis. A rotation angle and the scaled magnitude of the original vector will be the result of this vector operation (the x component of the result). The vectoring function based on the minimization of the y component of the residual vector at each rotation. The rotational direction is determined by the sign of the residual y component. At the end of the iterations the angle accumulator will contain the traversed angle if it is initialized with zero.

CORDIC is a good choice for hardware solutions such as FPGA in which cost minimization is more important than throughput maximization. In software implementations, it is a good solution for helping to conserve memory by enabling most of the code and data be shared between routines for trigonometric and hyperbolic functions.

Rotation of vector through specific angle has wide applications in robotics, digital signal processing, graphics, games, and animation. But, An optimized coordinate rotation digital computer algorithm for vector rotation through fixed and known angle is not available. Movements of robots are very often realized by successive rotations through small fixed angles and translations of the links. The translation operations are calculated by simple additions of coordinate values while the new coordinates of a rotational step could be accomplished by suitable successive rotations through a small fixed angle which could be performed by a

CORDIC circuit for fixed rotation. Similarly, interpolation of orientations between key-frames in computer graphics and animation could be performed by fixed CORDIC rotations. There are plenty of examples of uniform rotation starting from electrons inside an atom to the planets and satellites.

An animated mechanical clock is a simple example of uniform rotations which perform one degree rotation each time. Some applications like games, graphic, and animation where high-speed constant rotation is required. Constant rotations of an object are very often used in simulation, modelling, games, and animation. Using simple and dedicated CORDIC circuits efficient implementation of rotation through a known small angle to be used in these areas could be implemented efficiently. Similarly, CORDIC has lots of applications in the field of communication, signal processing and many other scientific and engineering applications where the multiplication of complex number with a known complex constant is often encountered.

An interesting feature of the CORDIC algorithm is that it uses a sequence of elementary rotations to realize a variety of complicated, nonlinear elementary functions. Each elementary rotation requires two simple simultaneous shift-and-add operations to implement. By unfolding the iterations for elementary rotation, a pipelined CORDIC array processor can be realized which will offer sustained high computing throughput rate. A reduction of the number elementary iterations for a specific algorithm, thus, will significantly reduce the latency (time between the availability of input data and available output), as well as hardware cost.

II. Cordic Fundamentals

The rotation of a given vector (xi, yi) is realized by means of a sequence of rotations with fixed angles which results in overall rotation through a given angle or result in a final angular argument of zero.

In this section, the computation of vector rotation illustrated in Fig. 1, which maps vector (x, y) to (x′, y′) according to the equations

$$x' = x \cos \Phi - y \sin \Phi \ldots(1)$$
$$y' = y \cos \Phi + x \sin \Phi \ldots (2)$$

Where Φ is a rotation angle. Here four multiplications and two additions are needed to compute x′ and y′ provided that the values of cos Φ and sin Φ are available.

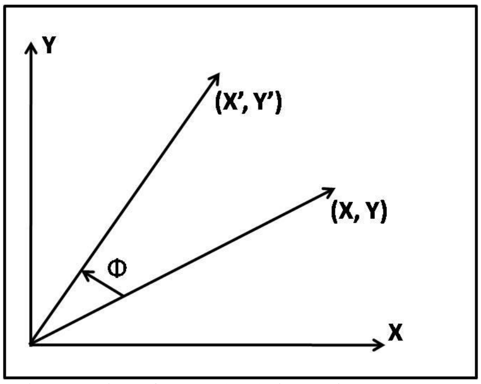

Fig1.Rotation of vector (x, y) by Φ degrees

In the following, (x_i, y_i) and (x_{i+1}, y_{i+1}) denotes the input and output to the rotation by $\Phi_i$respectively.

$$x_{i+1} = x_i\cos\Phi i - y_i \sin \Phi_i \ldots(3)$$
$$y_{i+1} = y_i\cos\Phi i + x_i \sin \Phi_i \ldots (4)$$

To proceed, let us assume that $-\pi/2 < \Phi i < \pi/2$. Using tan Φ = sin Φ / cos Φ, above equation can be rewritten as

$$x_{i+1} = \cos\Phi i(x_i - y_i \tan \Phi i)\ldots(5)$$
$$y_{i+1} = \cos\Phi i(y_i + x_i \tan \Phi i) \ldots (6)$$

The main idea of the CORDIC algorithm is that multiplication by tan Φi can be based on shifting, when tan $\Phi i = \pm 2^{-i}$,where i belongs to {0, 1, 2, . . . }.

Under this condition, the above equation becomes

$$x_{i+1} = \cos\Phi i(x_i - d_i \cdot y_i \cdot 2^{-i})\ldots(7)$$
$$y_{i+1} = \cos\Phi i(y_i + d_i \cdot x_i \cdot 2^{-i}) \ldots(8)$$

The CORDIC algorithm is usually performed in two ways with the number of iterations is fixed in the beginning:

1. Input a vector to be rotated by specific angle and output the resulting vector
2. Input a vector to be rotated onto the x axis and output the require rotation angle

Initialize z_i with the value of the desired rotation angle and then start the iterations of the algorithm using the following equations

$$x_{i+1} = x_i - y_i \cdot d_i \cdot 2^{-i} \ldots (9)$$
$$y_{i+1} = y_i + x_i \cdot d_i \cdot 2^{-i} \ldots (10)$$
$$z_{i+1} = z_i - d_i \cdot \tan^{-1}(2^{-i}) \; (11)$$

Where

$$d_i = \begin{cases} & \\ -1 & \text{otherwise} \end{cases}$$

These three equations are used for the vector rotation in CORDIC algorithm.

The calculations in CORDIC is performed by using only shifters, registers and adder / subtractors. Adder/ subtractors are used for the binary addition and subtraction. The single bit shifting performed by shift registers according to the algorithm. LUTs (look up tables) are used to set the value of the constants according to the demand of angle setting for the algorithm. For computation of sine and cosine different hardware is used in CORDIC. Here iterative rotations is based on a point around the origin on the x-y plane are considered. The coordinates of the rotated point and the remaining angle to be rotated are calculated In each rotation,

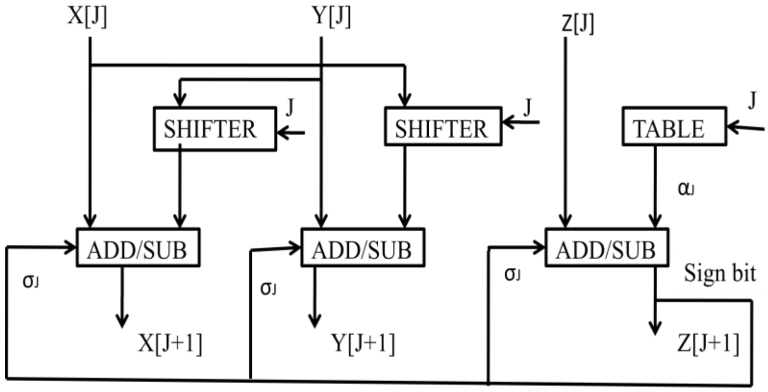

Fig. 2. Basic CORDIC circuit

Hardware implementation for CORDIC arithmetic requires three registers for x, y and z, two shifter to supply the terms $2^{-i}x$ and $2^{-i}y$ to the adder/subs tractor units and a look up table to store the values of $\alpha_i = \tan^{-1}2^{-i}$. The d_i factor (-1 and 1) selects the shift operand or its complement. The initial inputs to the architectures are $X_0=1$, $Y_0=0$. The structure requires a pre-processing unit to converge the input angles to the desired range and a post processing unit to fix the sign of outputs depending on the initial angle quadrants.

To obtain sine and cosine values of a given angle z_0, iterative structure takes the value of (x_0,y_0)as (1,0) in the first clock cycle. From the next clock cycle onwards it takes the feedback values and the operation continues till the required output is obtained. The control signal for the input registers is provided by a state-machine designed for the purpose. To get an N bit precise output, the structure requires iterating at least N times. Hence, it requires a minimum of N clock cycles for required output. A lookup table is an array that replaces runtime computation with a simpler array indexing operation. It is used to set the values of the constants according to the demand of angle setting for the algorithm.

Look up table is used reduce the computational complexity and the number of resources used.Angle calculation is done using look up table. A look up table containing the $\tan^{-1}(2^{-i})$ values. By using look up table, the number of iteration for the micro rotations can be reduced. Thus speed of the CORDIC can be improved. In this paper, a look table stores the values of $\tan^{-1}(2^{-i})$ so that in each iteration it can directly take the values from table and make calculation simpler and fast. A barrel shifter is a digital circuit that can be implemented as a sequence of multiplexers. The main function of a barrel shifter is to shift a data word by a specified number of bits in one clock cycle. Barrel shifter is commonly used in the hardware implementation of floating-point arithmetic. Here it is used for reducing the area and time complexity.

III. Implementation Procedure

CORDIC circuit implementation consists of three steps in development.

- Look up table
- Angle calculation
- Co-ordinate calculation

Here the three inputs given to the algorithm are $x_0,y_0,z_0.$ The input (x_0,y_0) is the input co-ordinate. If the shift of a given vector takes place from the initial position to a specific angle, initial co-ordinate will be the input to the co ordinate calculation. In this paper the initial co ordinate (x_0, y_0) is always taken as (1,0).The output co ordinate will be the final result after vector rotation through specific angle. The final output will get after less number of iteration. Here eight fixed number of iteration is performed to get the result. For any large given angle, the output will get with the same number of iteration. The input angle is takes its radian value and it again covert to its binary format. To increase the precision and accuracy, here the input is taken as 20-bit binary values. Here the first four digits represent the integer and the other digits represent the decimal values.

Angle calculation is based on the look up table values. So the look up table initialisation is done first before the angle calculation. The input angle z_i is given as the input for the angle calculation. The input angle is also taken as the 20 bit binary value. A look up table initialised for the storage of pre determined values of the angles. Angle calculation makes use of this stored value from the look up table in order to get the result within a small duration and thus increase the speed. So the use of tables reduces time complexities as well as increases the speed of the algorithm.

IV. Results And Discussions

VLSI implementation of modified CORDIC processor for specific angle of rotation based applications has been simulated using ModelSim. Synthesis of look up table, angle calculation, co ordinate calculation has been analysed by Cyclone-II. The timing analysis was carried out in Altera Quartus-II classic timing analyser tool. The power analysis of CORDIC was carried out in Altera Quartus-II Powerplay power Analyzer tool.

1. Synthesis Output Of Look Up Table: This system achieves 0.04% of total logical elements, 0.04% of total combinational elements, 0.00% of total dedicated registers and 10.94% of total available I/O pins. The RTL output of look up table in CORDIC system is shown in the fig3.

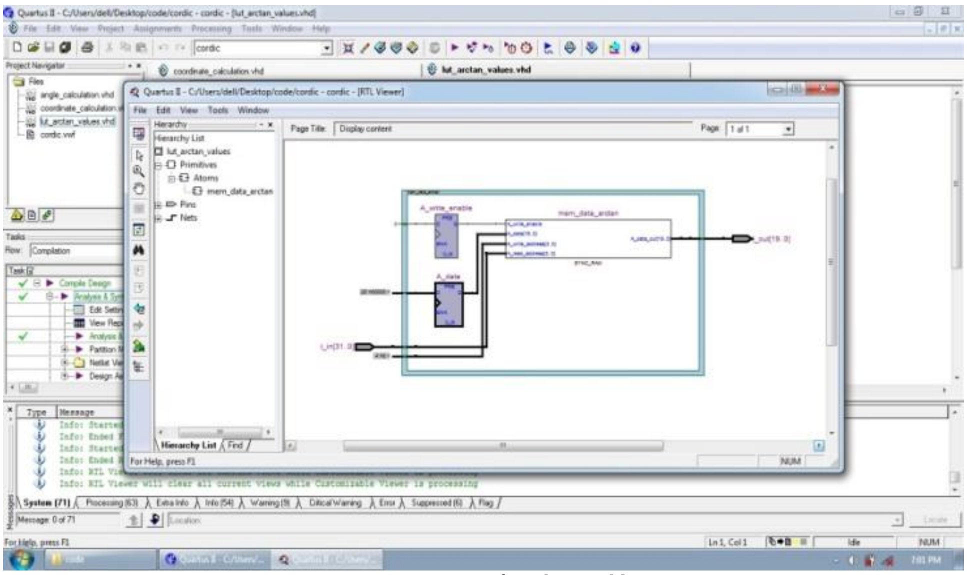

Fig 3.RTL Output of Look up Table

2. Analysis Output Of Look Up Table: The look up table of CORDIC system achieves worst case t_{pd}as12.027ns. The power analysis of look up table in CORDIC was carried out in Altera Quartus-II Powerplay power Analyzer tool. This system consumes a total thermal power of 115.77mW which includes core static thermal power dissipation of 79.94mW, core dynamic thermal power dissipation is 0.00Mw,core static thermal power dissipation is 79.94mW and I/O thermal power static dissipation of 35.83mW. The power analysis report for look up table in CORDIC is shown in the fig 4.

3. Simulation Output Of Look Up Table: Simulation of look up table in CORDIC was carried out in ModelSim. The input is given in i_in and the output is taken from i_out. i_in is the input which is asking for the number of iteration. Here total 9 iterations are taken so that i_in value between 0 to 8.Simulation output of look up table is shown in fig 5.

4. Synthesis Output Of Angle Calculation: This system achieves 0.71% of total logical elements,0.65% of total combinational elements, 0.31% of total dedicated registers and 9.2% of total available I/O pins. RTL output of Angle calculation is shown in fig 6.

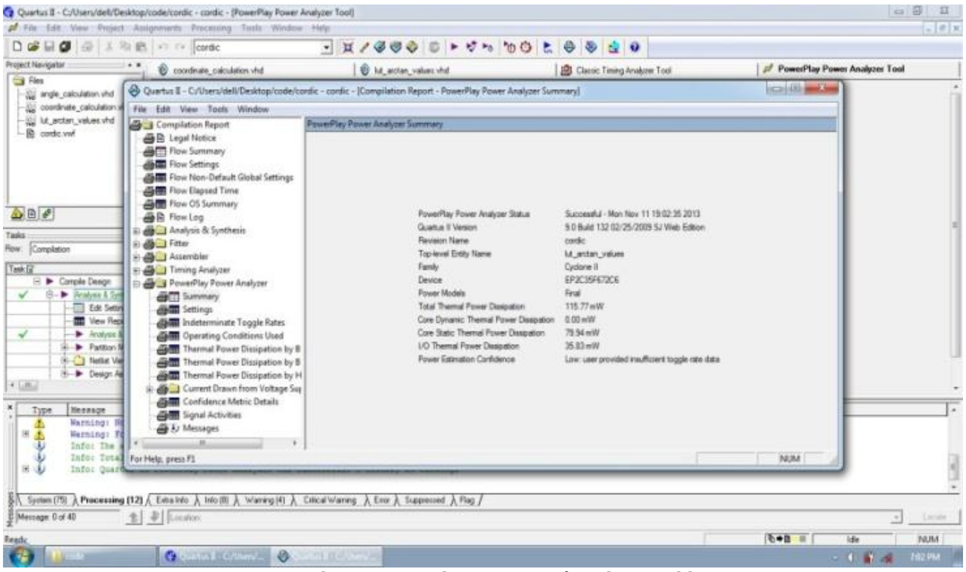

Fig 4.The Power Analysis Report of Look Up Table

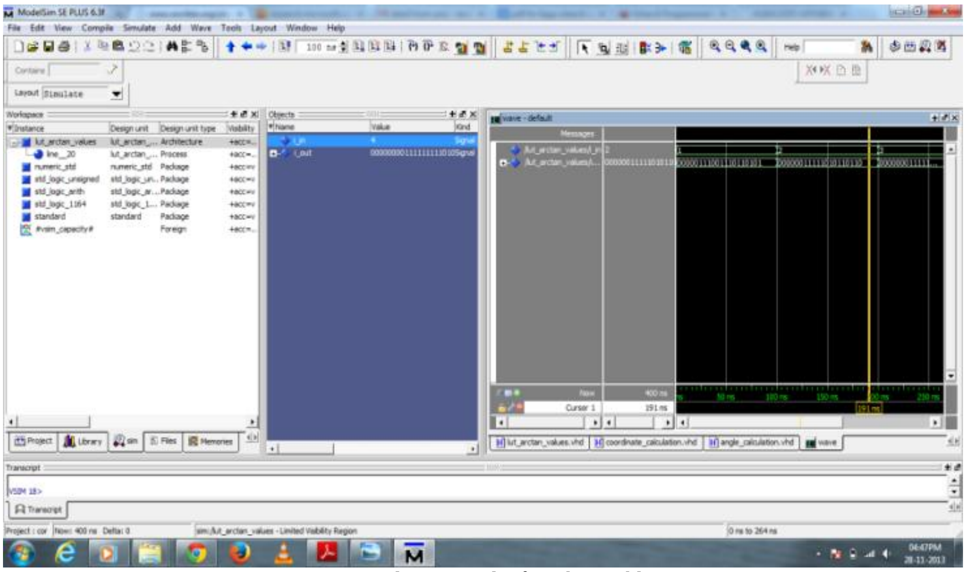

Fig 5.Simulation Result of Lookup Table

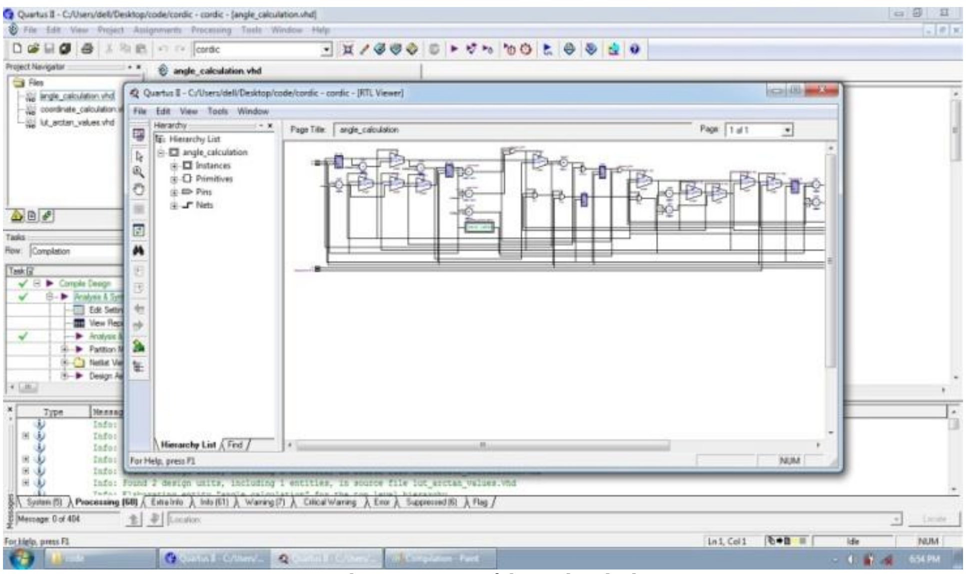

Fig 6. The RTL output of the angle calculation

5. Analysis Output of Angle Calculation: This system consumes a total thermal power of 115.05mW which includes core static thermal power dissipation of 79.94mW, core dynamic thermal power dissipation is 0.00Mw, core static thermal power dissipation is 79.94mW and I/O thermal power static dissipation of 35.11mW. The power analysis report for angle calculation in CORDIC is shown in the fig 7.

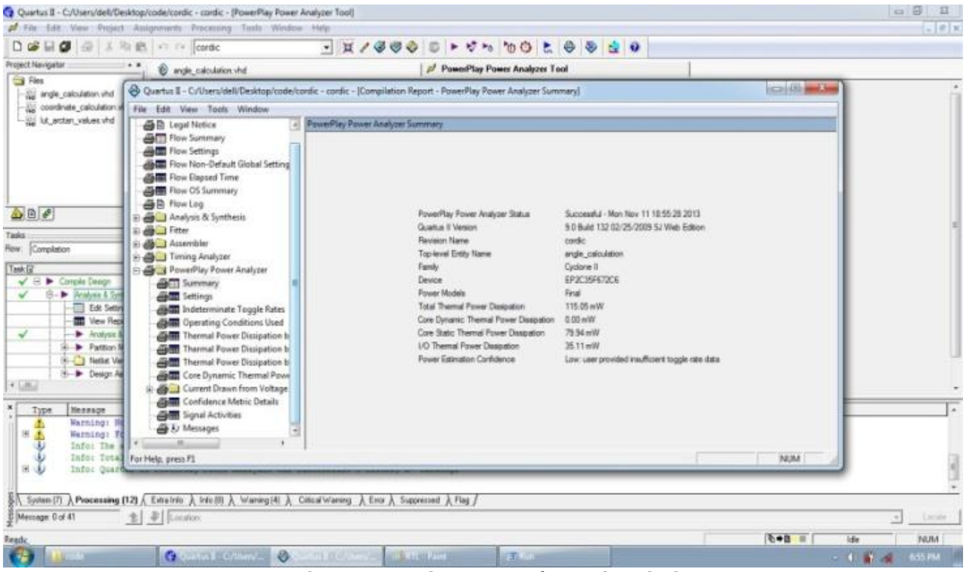

Fig 7. The Power Analysis Report for Angle Calculation

6. Simulation Output Of Angle Calculation: The angle calculation in CORDIC circuits was carried out in ModelSim and the waveforms are shown in the fig 8.The input angle is given in the form of 20 bit binary format in input angle. The clock pulse and reset signals are given through clk and reset pins respectively. Sign bit is changing according to the iteration. Output is taken from output angle.

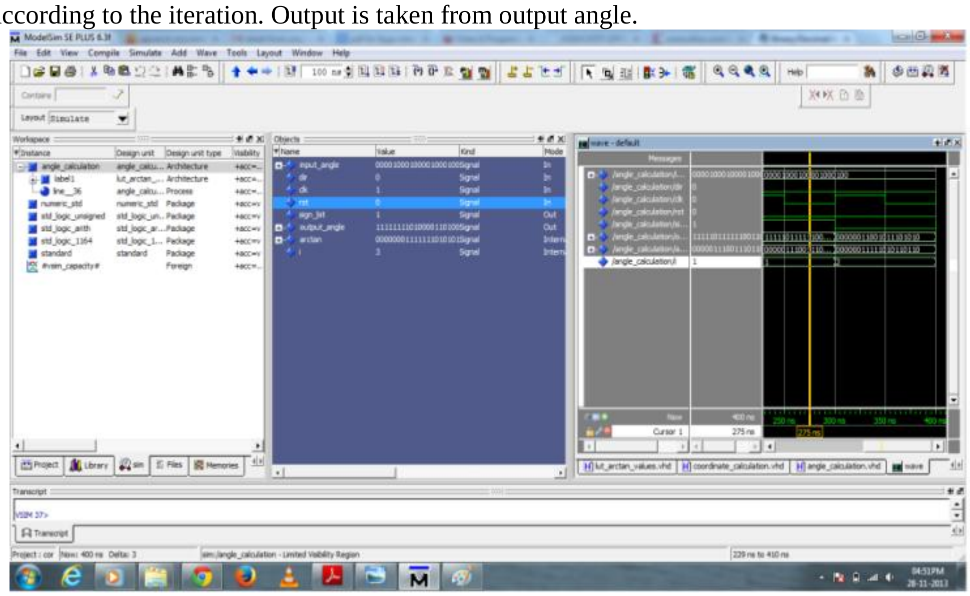

Fig 8. Simulation output of angle calculation

7. Synthesis Output of Co ordinate Calculation: The Co ordinate calculation in CORDIC structure was synthesized in Cyclone-II. This system achieves 1.39% of total logical elements, 1.28% of total combinational elements, 0.57% of total dedicated registers and 16.63% of total available I/O pins. The synthesis and compilation report of coordinate calculation in CORDIC system is shown in the fig 9.The RTL output of coordinate calculation in CORDIC system is shown in the fig 10.

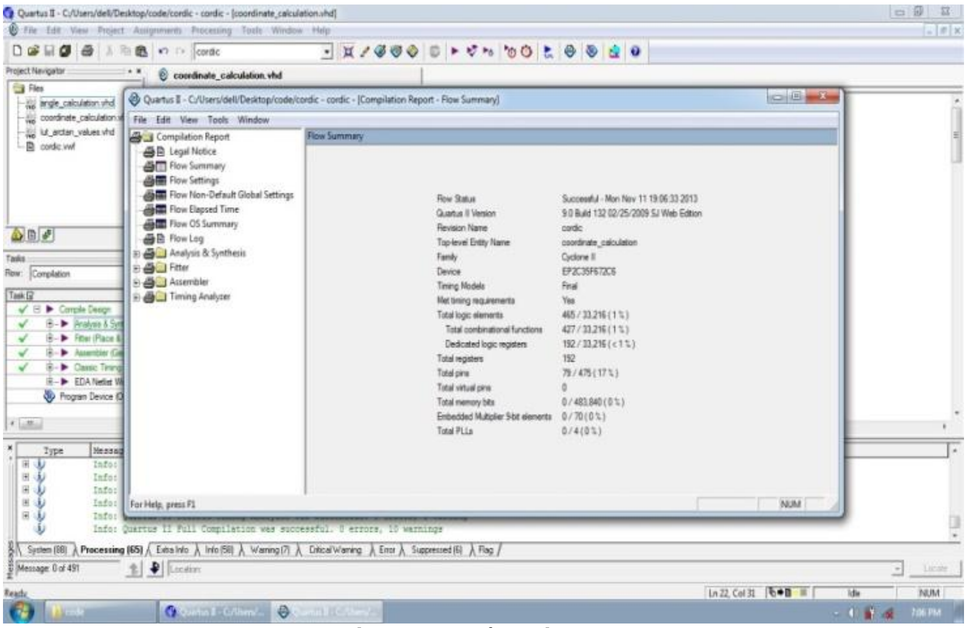

Fig 9.Compilation Report of Coordinate Measurement

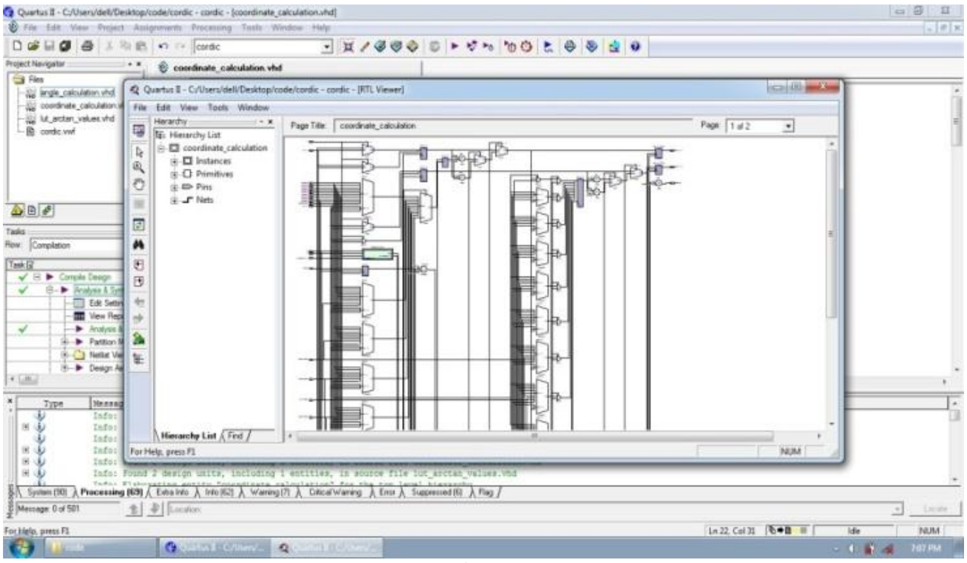

Fig 10. RTL Output of Coordinate Measurement

8. Analysis Output of Co ordinate Calculation: The timing analysis of co ordinate calculation in CORDIC system was carried out in Altera Quartus-II classic timing analyser tool. This system achieves a maximum clock set up time of 12.308ns (81.25MHz), worst case T_{su} of 12.326ns, worst case T_{co} of 8.373ns and worst case T_h of -3.365ns. The timing analysis report for coordinate calculation in CORDIC is shown in the fig 11

The power analysis of co ordinate calculation in CORDIC was carried out in Altera Quartus-II powerplay power analyser tool. This system consumes a total thermal power of 119.28mW which includes core static thermal power dissipation of 79.95mW, core dynamic static thermal power dissipation and I/O thermal power static dissipation of 39.33mW. The power analysis report for co ordinate calculation in CORDIC is shown in the fig 12.

9. *Simulation Output of Co ordinate Calculation:* Simulation of coordinate calculation was carried out in ModelSim. The input coordinates is given in the form of 9 bit binary format in $x_{in}$and y_{in}. The clock pulse and reset signals are given through clk and reset pins respectively. Sign bit is changing according to the iteration. Output is taken from $x_{out}$and y_{out}. Simulation output of coordinate calculation CORDIC is shown in fig 13

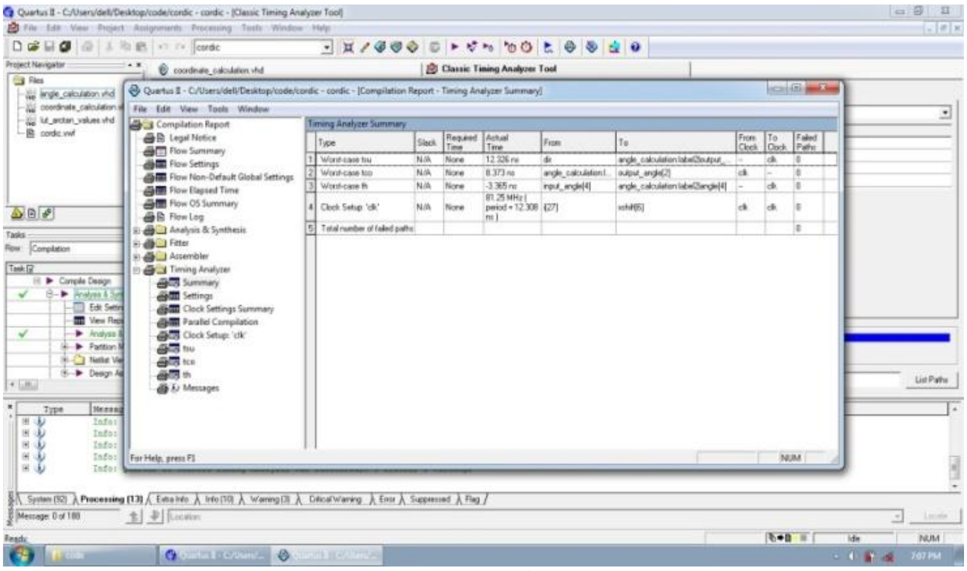

Fig 11.The Timing Analysis Report for Coordinate Calculation

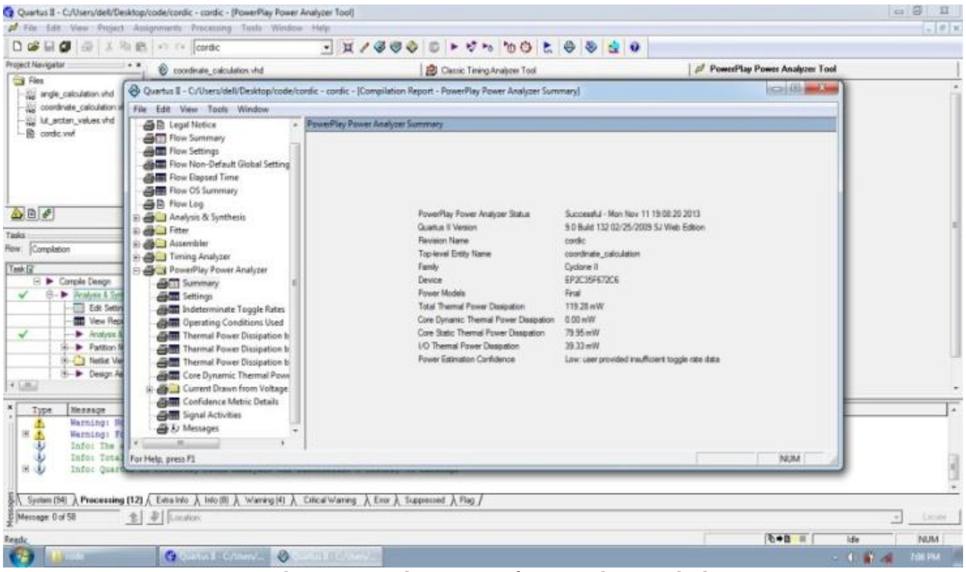

Fig 12.The Power Analysis Report for Co Ordinate Calculation

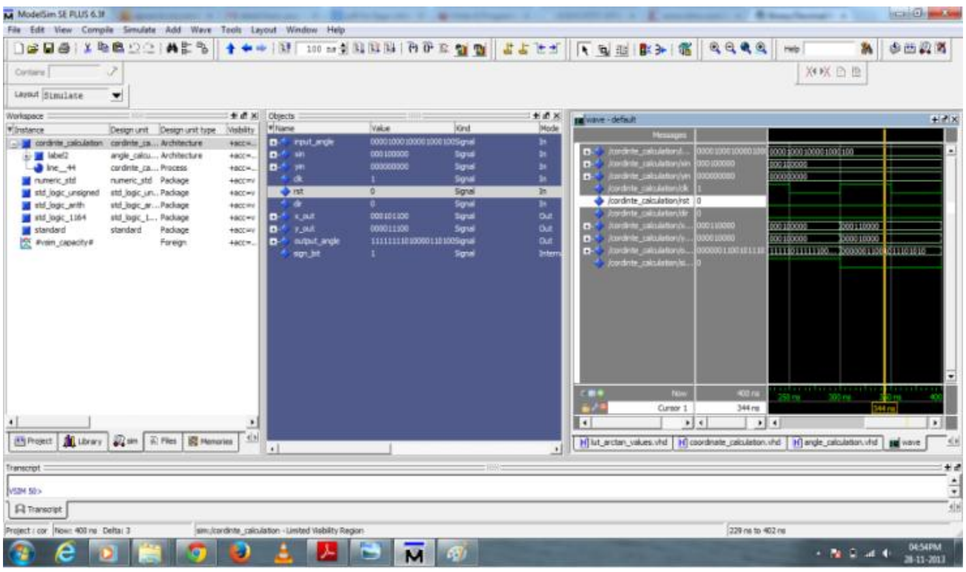

Fig 13. Simulation Output of Coordinate Calculation

V. Conclusion

CORDIC algorithm is used for the rotation of vectors through fixed and known angle. The basic concept of CORDIC is based on the two dimensional geometry. Here the inputs given to the CORDIC circuits are the initial co ordinate of a vector and the fixed angle that is to be rotated. The input given to the circuit is thus divided into two parts. In the first part, Angle calculation, the input will be the specific angle which is to be rotated. For the simple and fast calculation a look up table is initialized first before the angle calculation. Look up table consists of the pre determined values of the angles. In the second part, Coordinate calculation, input will be the initial coordinates of the given vector. The calculation of final coordinates after complete iteration is based on the fixed and known angle. To reduce the area and time complexities pre shifting schemes in barrel shifters is adopted in the coordinate calculation. Here the output will be the final co ordinates and the specified angle after the rotation through the given angle. The future work is to reduce the utilization of excess resources and thus reduce the computational time.

References

[1] Hu .Y. H. andChern.H. H.M,(1996),'A novel implementation of CORDIC algorithm using Backward angle recoding (BAR)', IEEE Trans.Comput., vol. 45, no. 12, pp.1370-1378

[2] Hu. Y. H. and Naganathan.S,(1993), 'An angle recoding method for CORDIC algorithm implementation,'IEEE Trans. Comput., vol.42, no. 1, pp. 99–102

[3] Lin.C. H. and Wu. A. Y, (2005), 'Mixed-scaling-rotation CORDIC (MSRCORDIC) Algorithm and architecture for high-performance vector rotational DSP applications,' IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 52, no. 11, pp. 2385–2396.

[4] Maharatna.K.,Banerjee.S, Grass.E, Krstic. M, and TroyaA. (2005), 'Modified virtually Scaling free adaptive CORDIC rotator algorithm and architecture,'IEEETrans.Circuits Syst. forVideo Technol., vol. 15, no.11, pp. 1463–1474

[5] Meher. P. K, Valls. J, Juang T.-B., Sridharan.K, and Maharatna.K,(2009), '50 years of CORDIC: Algorithms, architectures and applications,'IEEETrans. Circuits Syst. I, Reg. Papers, vol. 56, no. 9, pp. 1893–1907.

[6] Vachhani. L, Sridharan. K., and Meher. P. K,(2009),'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

[7] Volder.J. E, (1959),'The CORDIC trigonometric computing technique,' IRE Trans. Electron. Comput., vol. EC-8, pp. 330–334.

[8] Walther. J. S, (1971), 'A unified algorithm for elementary functions,' in Proc. 38thSpring JointComput. Conf., pp. 379–385.

[9] Wu.C.-S, Wu .A.-Y, and Lin. C.-H,(2003), 'A high-performance/low-latency vector rotational CORDIC architecture based on extended elementary angle set and trellis- based searching schemes,' IEEE Trans. CircuitsSyst. II, Analog Digit. Signal Process, vol.50, no. 9, pp. 589–601.

[10] Freeman.S and O'Donnell.M, (1995) 'A complex arithmetic digital signal processor using cordic rotators', in Proc. IEEE Int. Conf. Acoust. Speech, Signal Process. (ICASSP),pp. 3191–3194.