Abatement of Area and Power In Multiplexer Based on Cordic Using Cadence Tool

Uma P.[1], AnuVidhya G.[2]

VLSI Design, Karpaga Vinayaga College of Engineering and Technolog, G.S.T Road,Chinna Kolambakkam-603308

***Abstract* – CORDIC is an iterative Algorithm to perform a wide range of functions including vector rotations, certain trigonometric, hyperbolic, linear and logarithmic functions. Both non pipelined and 2 level pipelined CORDIC with 8 stages, using two schemes was performed. First scheme was original unrolled CORDIC and second scheme was MUX based pipelined unrolled CORDIC. Compared to first scheme, the second scheme is more reliable, since the second scheme uses multiplexer and registers. By adding multiplexer the area is reduced comparatively to the first architecture, since the first scheme uses only addition, subtraction and shifting operation in all the 8 stages.8 iterations are performed and it is implemented on QUARTUS II software. For future work, the number of iterations can be increased and also increase the bit size. This can be implemented in (digital) CADENCE software.**

***Keywords:* CORDIC, rotation mode, multiplexer, pipelining, QUARTUS.**

I. INTRODUCTION

The CORDIC is a class of hardware-efficient algorithms for the computation of trigonometric and other transcendental functions that use only shifts and adds to perform. The CORDIC set of algorithms for the computation of trigonometric functions was developed by Jack E. Volder in 1959 to help in building a real-time navigational system for the B-58 supersonic bomber. Later, J. Walther in 1971 extended the CORDIC scheme to other transcendental functions. The CORDIC method of functional computation is used by most handheld calculators (such as the ones by Texas Instruments and Hewlett-Packard) to approximate the standard transcendental functions.

Calculators can only perform four operations inexpensively:

1. Addition and Subtraction
2. Storing in memory and Retrieving from memory
3. Digit shift (multiplication/division by the base)
4. Comparisons

The CORDIC Algorithm is a unified computational scheme to perform

1. Computations of the trigonometric functions: sin, cos and arctan.
2. Computations of the hyperbolic trigonometric functions: sinh, cosh and arctanh.
3. It also compute the exponential function, the natural logarithm and the square root.
4. Multiplication and division.

CORDIC revolves around the idea of "rotating" the phase of a complex number, by multiplying it by a succession of constant values.

However, the "multiplies" can all be powers of 2, so in binary arithmetic they can be done using just shifts and adds; no actual "multiplier" is needed.

Both non pipelined and 2 level pipelined CORDIC with 8 stages and using two schemes was done. First scheme using adders in all the stages and second scheme using multiplexers only in the second and third stages, the other stages are as same as first scheme. The second scheme achieves less area compared to original unrolled CORDIC (first scheme). It is performed in QUARTUS II. Multiplexer has been proposed for the ASIC implementation of unrolled CORDIC (Coordinate Rotation Digital Computer) processor

II. CORDIC ALGORITHM

The CORDIC algorithm is an iterative method of performing vector rotations by arbitrary angles using shifts and addition. In the rotation mode, CORDIC may be used for converting a vector in polar form to rectangular form. In the vector mode, it converts a vector in rectangular form to polar form. Both the modes are derived from the general rotation transform.

$$X_{fin} = X_{in}\cos\theta\text{-}Y_{in}\sin\theta \qquad (1)$$
$$Y_{fin}=X_{in}\sin\theta+Y_{in}\cos\theta \qquad (2)$$

Cartesian plane by an angle θ to another vector with the coordinates. The rotation may be achieved by performing a series of successively smaller elementary rotations $\theta_1,\theta_2,\theta_3\ldots.\theta_N$. Rotation of the vector by an angle can be rewritten as

$$X_{i+1}=X_i\cos\theta\, i\text{-}Y_i\sin\theta\text{i} \qquad (3)$$
$$Y_{i+1}=X_i\sin\theta\text{i}+Y_i\cos\theta\text{i} \qquad (4)$$
$$X_{i+1}/\cos\theta\text{i}=X_i\text{-}Y_i\tan\theta\text{ i} \qquad (5)$$
$$Y_{i+1}/\cos\theta\text{i}=Y_i+X_i\tan\theta\text{i} \qquad (6)$$

The computational complexity of (5), (6) can be reduced by rewriting these equations as

$$X_{i+1}=X_i\text{-}Y_i\tan\theta\text{i} \qquad (7)$$
$$Y_{i+1}=Y_i+X_i\tan\theta\text{i} \qquad (8)$$
$$(X_{fin},Y_{fin}) = (\,x_N/\pi_0^N\cos\theta\text{i}\,,y_N/\pi_0^N\cos\theta\text{i}\,) \qquad (9)$$

To get the final coordinate values, perform division (x_N,y_N) by $\prod_i^N cos\,\theta_i$. The value of θ_i for i = 0, 1, 2…N is chosen such that $tan_{\theta i}$ is 2^{-i}. This reduces the multiplication by the tan to simple shift operation. As the iteration increases, θ_i becomes smaller and smaller.

Terminate the iteration when the difference between θ = $\sum_0^N\theta$i becomes very small for some value of N.

The remaining angle by which the vector needs to be rotated after completion of i iterations is indicated by the parameter zi+1 and is defined by.

$$Z_{i+1}=Z_i\text{-}\theta_i \qquad (10)$$

θ_i is considered to be positive when the rotation required is anticlockwise and is negative otherwise. The direction of this rotation depends on the δi.

$$\delta_i=sgn(Z_i) \qquad (11)$$

The computation of $\pi_0^N\cos\theta$i may be simplified as follows: Since cos θi=1 for very smaller values of θ_i , $\pi_0^N\cos\theta$i may be computed for N=8 and may be used for any value of N>8.

III. PREVIOUS WORK

THE UNROLLED CORDIC IN ROTATION MODE

In rotation mode, CORDIC can simultaneously compute the sine and cosine of the input angles. In this mode, set the y component of the input vector to zero, x component to 1/k and the angle accumulator is initialized with the desired rotation angle θ. The output of angle accumulator decreases or increases depending on the most significant bit of the output of the previous stage. For rotation mode, the CORDIC equations are given by

$$X_{i+1}=X_i\text{-}Y_i.\delta_i.2^{-i} \qquad (12)$$
$$Y_{i+1}=Y_i+X_i.\delta_i.2^{-i} \qquad (13)$$
$$Z_{i+1}=Z_i\text{-}\delta_i.\tan^{-1}2^{\text{-i}} \qquad (14)$$
$$k=\prod_0^N cos\,\theta_i \qquad (15)$$

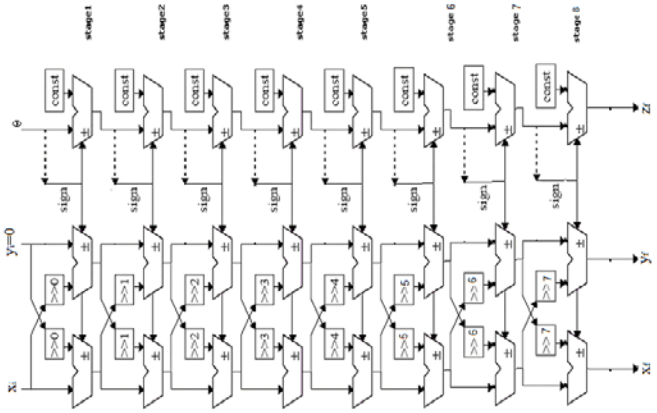


Fig. The Unrolled CORDIC

The architecture of the eight stage unrolled CORDIC is shown ; this consists of only adders, subtractors and shifters; accuracy improves as the number of stages increases. Addition or subtraction on the angle value takes place in each rotation of the vector depending on the most significant bit of previous angle. Perform division just by doing right shift using shift registers. This has the advantage of not using extra hardware for division and it results in less hardware complexity. Initially, assign constant values to x and y. These values are shifted by j bits, where j is the integer {0, 1, 2, 3, 4, 5, 6, 7} which results in division of x and y by 1, 2, 4, 8, 16, 32, 64 and 128 for every stage. In this mode, the vector is iteratively rotated to make new vectors in the intermediate stages to get the desired angle.

IV. PROPOSED METHODOLOGY

A. MUX BASED CORDIC

The scheme for reducing the area of the CORDIC using multiplexer is proposed for the ASIC implementation. This is adopted for the QUARTUS II based implementation . The area is reduced by removing some of the stages .

The first stage output of original unrolled CORDIC architecture is equal to xi, therefore we can directly write the output of first stage as

$$Y_1 = X_i \qquad (16)$$

$$X_1 = X_i \qquad (17)$$

If the first stage output is positive, then

$$Y_2 = Y_1 - \frac{x1}{2} = \frac{xi}{2} \qquad (18)$$

$$X_2 = X_1 + \frac{y1}{2} = 3Xxi/_2 \qquad (19)$$

The vector coordinates corresponding to negative output is

$$Y_2 = Y_1 + \frac{x1}{2} = 3Xxi/_2 \qquad (20)$$

$$X_2 = X_1 - \frac{y1}{2} = \frac{xi}{2} \qquad (21)$$

The output of the second stage is fixed. So implement the second stage using two Mux and choosing select line as the MSB bit of the previous angle accumulator output.

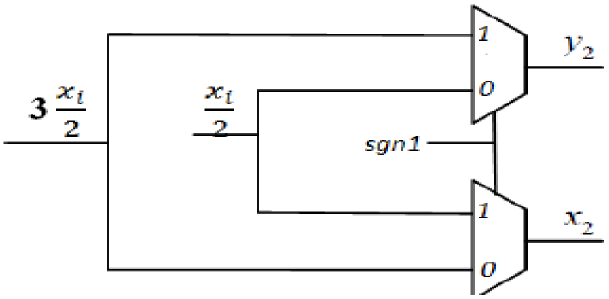


Fig. Mux Based CORDIC

To reduce the area, we replace the third stage with Mux. Since the third stage output also depends only on xi.

The block diagram of the CORDIC when the adders till third stage are replaced with Mux is shown . As the adders are replaced with Mux, the area of the circuit is reduced till 3rd stage. But the replacement of adders with Mux beyond the third stage results in an exponential increase in the number of Mux as shown in Table I.

$$Y_3 = Y_2 + \frac{X2}{4} = \frac{3Xxi}{2} + \frac{xi}{8} = \frac{13Xxi}{8} \qquad (22)$$

$$X_3 = X_2 - \frac{Y2}{4} = \frac{xi}{2} - \frac{3Xxi}{8} = \frac{xi}{8} \qquad (23)$$

For $sgn_1 = 0, sgn_2 = 0$

$$Y_3 = Y_2 + \frac{x2}{2} = \frac{xi}{2} + \frac{3Xxi}{8} = \frac{7Xxi}{8} \qquad (24)$$

$$X_3 = X_2 - \frac{y2}{4} = \frac{3Xxi}{2} - \frac{xi}{8} = \frac{11Xxi}{8} \qquad (25)$$

For $sgn_1 = 1, sgn_2 = 0$

$$Y_3 = Y_2 - \frac{x2}{4} = \frac{3Xxi}{2} - \frac{xi}{8} = \frac{11Xxi}{8} \qquad (26)$$

$$X_3 = X_2 + \frac{y2}{4} = \frac{xi}{2} + \frac{3Xxi}{8} = \frac{7Xxi}{8} \qquad (27)$$

For $sgn_1 = 0, sgn_2 = 1$

$$Y_3 = Y_2 - \frac{x2}{4} = \frac{xi}{2} - \frac{3Xxi}{8} = \frac{xi}{8} \qquad (28)$$

$$X_3 = X_2 + \frac{y2}{4} = \frac{3Xxi}{2} + \frac{xi}{8} = \frac{13Xxi}{8} \qquad (29)$$

For $sgn_1 = 1, sgn_2 = 1$

No. of eliminated stages	No. of Mux Required
1	0
2	2
3	6
4	14
5	30

Table1. Multiplexers required for eliminating different stages

B.PIPELINED MUX BASED UNROLLED CORDIC

The pipelined CORDIC use registers in between each iteration stage as shown. The advantage of pipelined unrolled CORDIC over the unrolled CORDIC is its higher frequency of operation. This property can be used in high speed applications. The number of registers depends on the number of stages in pipelining and there will be an increase in area. The first output of an N-stage pipelined CORDIC core is obtained after N clock cycles. Thereafter, outputs will be generated during every clock cycle. In this paper, pipelined registers are placed after fourth and seventh stages. Mux based pipeline unrolled CORDIC architecture in which pipeline registers are inserted at the output.

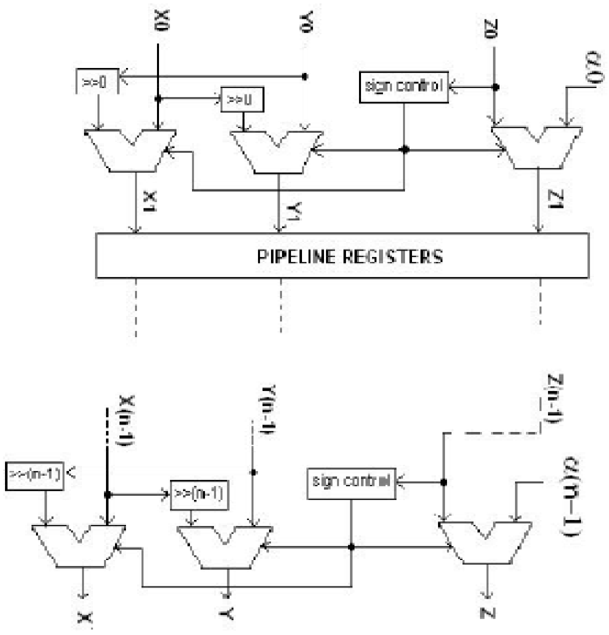


Fig. Pipelined CORDIC Using Registers

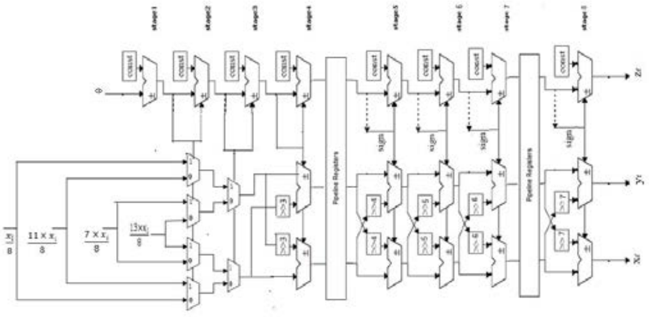


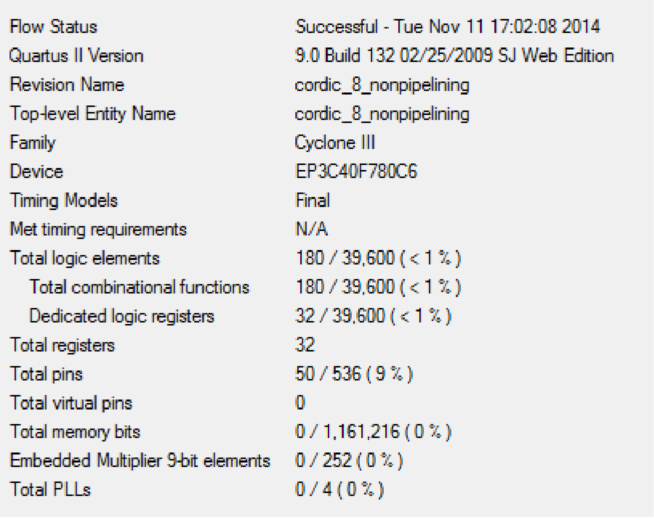
Fig. Pipelined MUX Based Unrolled CORDIC

V. SIMULATION/EXPERIMENTAL RESULTS

I. Results for Unrolled CORDIC

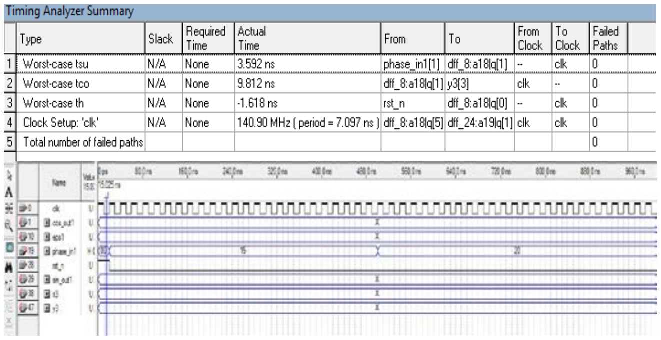
The Report implies the QUARTUS II Version, Top-level entity name, Family and Device models. The Device models used here is EP3C40F780C6 and it belongs to family Cyclone III or II.

The total pins, registers, logic elements and combinational functions used in the circuits are analysed. From this report information about the interior performance of the circuits are known



Flow Status	Successful - Tue Nov 11 17:02:08 2014
Quartus II Version	9.0 Build 132 02/25/2009 SJ Web Edition
Revision Name	cordic_8_nonpipelining
Top-level Entity Name	cordic_8_nonpipelining
Family	Cyclone III
Device	EP3C40F780C6
Timing Models	Final
Met timing requirements	N/A
Total logic elements	180 / 39,600 (< 1 %)
Total combinational functions	180 / 39,600 (< 1 %)
Dedicated logic registers	32 / 39,600 (< 1 %)
Total registers	32
Total pins	50 / 536 (9 %)
Total virtual pins	0
Total memory bits	0 / 1,161,216 (0 %)
Embedded Multiplier 9-bit elements	0 / 252 (0 %)
Total PLLs	0 / 4 (0 %)

Fig. QUARTUS Report



Timing Analyzer Summary

	Type	Slack	Required Time	Actual Time	From	To	From Clock	To Clock	Failed Paths
1	Worst-case tsu	N/A	None	3.592 ns	phase_in1[1]	dff_8:a18|q[1]	--	clk	0
2	Worst-case tco	N/A	None	9.812 ns	dff_8:a18|q[1]	y3[3]	clk	--	0
3	Worst-case th	N/A	None	-1.618 ns	rst_n	dff_8:a18|q[0]	--	clk	0
4	Clock Setup: 'clk'	N/A	None	140.90 MHz (period = 7.097 ns)	dff_8:a18|q[5]	dff_24:a19|q[1]	clk	clk	0
5	Total number of failed paths								0

Fig Timing Analyzer Summary Waveform

The waveform shows the timing Analyzer summary of the non-pipelined Cordic circuit. The details of the required time, actual time and clk period are calculated and also the total no. of failed path in the circuits are known .

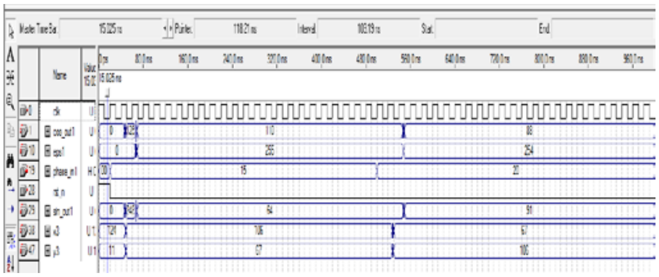


Fig .The Original Unrolled CORDIC Output Waveform

The Time Quest analyzer provides an intuitive and easy-to-use GUI that allows you to constrain and analyze designs efficiently. GUI has four planes. Each plane provide features that enhance the productivity of performing static timing analysis in the Time Quest analyzer.

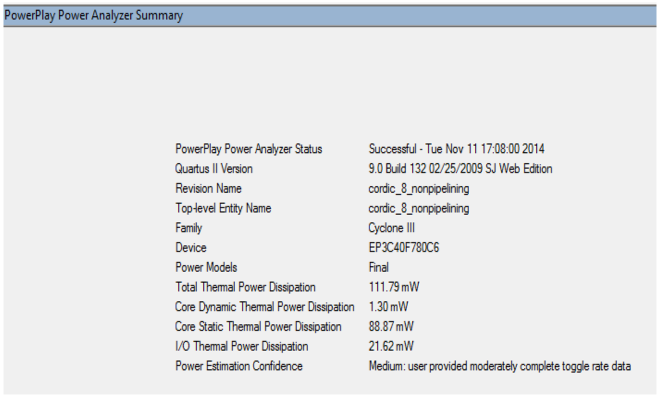
The output waveform shows the values of trigonometric functions for different angle values. Phase_in1, rst_n and clk are the inputs assigned to the block diagram. Cos_out1 , sin_out1 and eps1 are the outputs obtained. For different angle value the corresponding sin and cos values are calculated. For 30° corresponding hexadecimal value is 15 and output obtained for sin_out1 is 64 and cos_out 1 is 255.

For 45° corresponding value is 20 and output obtained for sin_out1 is 51 and cos_out1 is 254. Mainly the CORDIC algorithm is preferred to calculate the trigonometric values for different angle value.

INPUT	INPUT	OUTPUT	OUTPUT
ANGLE VALUE	CALCULATED VALUE	SIN_out1 VALUE	COS_out1 VALUE
30°	15	64	255
45°	20	51	254

TABLE 2. CORDIC Waveform Calculation

POWER ANALYZER



PowerPlay Power Analyzer Summary

PowerPlay Power Analyzer Status	Successful - Tue Nov 11 17:08:00 2014
Quartus II Version	9.0 Build 132 02/25/2009 SJ Web Edition
Revision Name	cordic_8_nonpipelining
Top-level Entity Name	cordic_8_nonpipelining
Family	Cyclone III
Device	EP3C40F780C6
Power Models	Final
Total Thermal Power Dissipation	111.79 mW
Core Dynamic Thermal Power Dissipation	1.30 mW
Core Static Thermal Power Dissipation	88.87 mW
I/O Thermal Power Dissipation	21.62 mW
Power Estimation Confidence	Medium: user provided moderately complete toggle rate data

Fig . Power Analyzer Report

The power play power analyzer performs the post fitting power analysis and produces a power report that highlights the block type, entity and power consumed.

Power analyzer report says the total power thermal dissipation, core dynamic thermal power dissipation, core static thermal power dissipation and input /output power dissipation. From this report the static and dynamic power rate and also input/output power rate are analysed.

II. Results for Pipelined MUX Based Unrolled Cordic

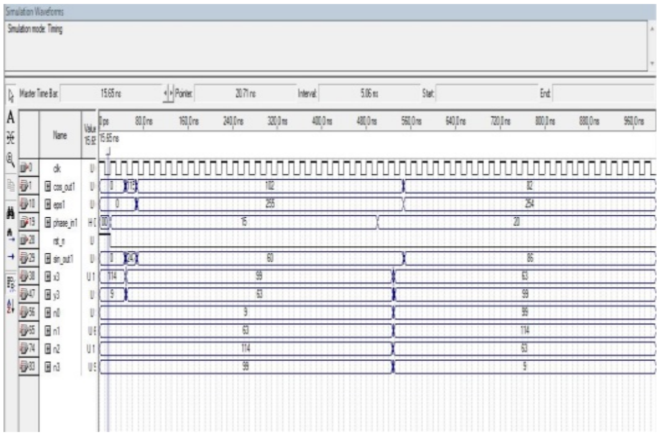


Fig. Output Waveform Of Pipelined MUX Based Unrolled Cordic

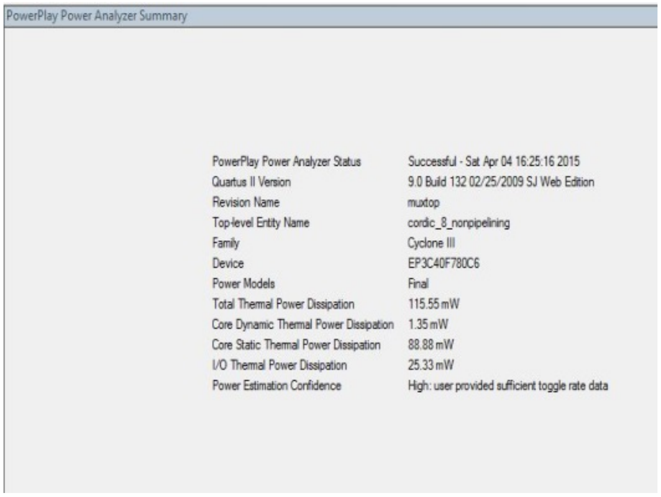
The output waveform shows the values of trigonometric functions for different angle values. Phase_in1, rst_n and clk are the inputs assigned to the block diagram. Cos_out1 , sin_out1 and eps1 are the outputs obtained. For different angle value the corresponding sin and cos values are calculated. For 30° corresponding hexadecimal value is 15 and output obtained for sin_out1 is 247 and cos_out 1 is 115.

For 45° corresponding value is 20 and output obtained for sin_out1 is 86 and cos_out1 is 82. Mainly the CORDIC algorithm is preferred to calculate the trigonometric values for different angle value.

INPUT	INPUT	OUTPUT	OUTPUT
ANGLE VALUE	MEASURED VALUE	SIN_out1 VALUE	COS_out1 VALUE
30°	15	247	115
45°	20	86	82

TABLE 3.Output Waveform Calculation

POWER ANALYZER



PowerPlay Power Analyzer Summary

PowerPlay Power Analyzer Status	Successful - Sat Apr 04 16:25:16 2015
Quartus II Version	9.0 Build 132 02/25/2009 SJ Web Edition
Revision Name	muxtop
Top-level Entity Name	cordic_8_nonpipelining
Family	Cyclone III
Device	EP3C40F780C6
Power Models	Final
Total Thermal Power Dissipation	115.55 mW
Core Dynamic Thermal Power Dissipation	1.35 mW
Core Static Thermal Power Dissipation	88.88 mW
I/O Thermal Power Dissipation	25.33 mW
Power Estimation Confidence	High: user provided sufficient toggle rate data

Fig. Power Analyzer Report

VI. CONCLUSION AND FUTURE WORK

CORDIC algorithm was used to find out the trigonometric, hyperbolic, linear and logarithmic functions. In CORDIC algorithm two schemes was discussed .First scheme was original unrolled CORDIC and second scheme was MUX based pipelined unrolled CORDIC. Compared to first scheme, the second scheme is more reliable, since the second scheme uses multiplexer and registers. By adding multiplexer the area is reduced comparatively to the first architecture, since the first scheme uses only addition, subtraction and shifting operation in all the 8 stages. 8 iterations are performed and it is implemented on QUARTUS II software.

For future work, the number of iterations can be increased and also increase the bit size. Thus can be implemented in (digital) CADENCE software. The Quartus results are compared with the Cadence results for future work.

REFERENCE

[1] J.E. Volder, “The CORDIC Trigonometric Computing Technique”, IEEE Transactions on Electronic computer, vol. EC-8, pp. 330-334, 1959.

[2] J. Walther, “a unified algorithm for elementary functions,” proc. Spring joint comp. con & vol.38, pp.379-385, 1971.

[3] Vankka J; Kosunen M; Hubach J; Halonen K; "A CORDIC-based multicarrier QAM modulator," Global Telecommunications conference,1999.GLOBECOM’ 99,vol.1A,no.,pp. 173- 177vol.1a,1999 .

[4] Chen A; Mc Danell R; Boytim M; Pogue R; "Modified CORDIC demodulator implementation for digital IF-sampled receiver," Global Telecommunications Conference, 1995. GLOBECOM '95., IEEE , vol.2, no., pp.1450-1454 vol.2, 14-16 Nov 1995.

[5] Deprettere E; Dewilde P; Udo R; "Pipelined cordic architectures for fast VLSI filtering and array processing," Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '84. , vol.9, no., pp. 250- 253, Mar 1984.

[6] Supriya Aggarwal, Pramod K. Meher, and Kavita Khare “Area-Time Efficient Scaling-Free CORDIC Using Generalized Micro-Rotation Selection”

[7]. Terence K. Rodrigues and Earl E. Swartzlander, “Adaptive CORDIC: Using parallel Angle Recoding to Accelerate”,Proc.IEEE Transactions on computers,Vol.59,No.4,pp.522-531,2010.

[8]. R.Ranga Teja, P.Sudhakara Reddy,”SINE/COSINE Generator Using Pipelined CORDIC processor”,Proc. IACSIT International Journal of Engineering and Tecnology,Vol3,No.4,pp.431-434,2011

[9]. HerberiDawid, Heinrich Meyr, ”VLSI Implementation of the CORDIC algorithm using redundant arithmetic”,Proc.IEEE International Symposium on Circuits and Systems,Vol.3,pp.1089-1092,1992

[10]. Xueming Wei;Shengyuan Zhou, “A Novel Circuit Design Based CORDIC for QAM Modulator,”Communication, Circuits and Systems 2007.ICCCas 2007.

[11]. Peter Nilsson, “Complexity reduction in unrolled CORDIC architectures” “Electronics, circuits and system,2009.ICECS.