

Implementation of Area efficient CORDIC for QPSK Modulation

I. Sharath Chandra* and Joseph Beatrice Seventline

GITAM, Gandhi Nagar, Rushikonda, Visakhapatnam-530045, Andhra Pradesh, India.

Received 4 November 2017; Accepted 7 December 2017

Abstract

In this paper, we have designed a CORDIC algorithm, which is used to minimize the angle with help of several rotations. In the process, we have used the proposed CORDIC algorithm for simulating a QPSK modulator and its output is plotted using Modelsim tool. The main idea of the proposed system is to replace the normal adders with Low area carry select adders (CSLA). This CSLA can achieve fast arithmetic operation in various data processing techniques. Finally, comparison of various performance parameters like area, power and delay will be tabulated in comparison of proposed algorithm with the existing one.

Keywords: CORDIC algorithm, QPSK modulator, CSLA

1. Introduction

In today's technology, Digital Signal processor (DSP) is one of the important processor to speed up the operations and obtain error free results in the hardware implementation. The multipliers and adders are the most significant hardware units to perform operations such as filtering, convolution and compression, etc. To boost up the operation of the DSP units, J.E.Volder introduced an efficient CORDIC algorithm in the year 1959. With this algorithm, trigonometric functions can be computed very precisely with minimum hardware and latency. The CORDIC digital computer has special kind of digital units such as shifters, adders and subtractors for above-mentioned operations [1].

Over the last five decades, there were many improvements made in the conventional CORDIC algorithm based upon the requirements, where the latency, throughput, the scaling Factor, performance parameters like area, power were the main concern.

A novel redundant CORDIC algorithm for computation of sine and cosine functions was proposed later, in which the number of rotations performed for each calculation of trigonometric angle is a constant, in turn making the scale factor a constant, which is independent of the operand. This algorithm has eliminated the need of computation of the scale factor [2]. An enhanced Scaling-Free CORDIC algorithm was also introduced for the DSP applications to obtain better results in the implementation. This improved scaling free CORDIC algorithm was implemented on a microcontroller in comparison to the conventional CORDIC which resulted in high speed and accurate computations [3].

An optimized CORDIC algorithm is the one, which conquer the restrictive relation of the power, area, speed and precision in conventional CORDIC. Here optimization of the conventional CORDIC algorithm is done by considering the

above mentioned parameters and implemented on the FPGA [4]. Another memory less CORDIC algorithm has been introduced for the FFT computation [5]. With combination of radix and CORDIC algorithm, FFT processor design was implemented effectively to reduce the number of addition and multiplication operations. Also complexity of the twiddle factor in the butterfly computation of the FFT processor was minimized [6]. A Constant Coefficient quaternion multiplier is implemented with CORDIC algorithm for simple and less area requirements, but here, more sparse iterations were required to reach the good accurate results[7].

A Folded Pipelined Architecture for FFT with CORDIC Algorithm has minimized the number of functional units. In the folding operation, several butterfly units were present in the same column, which can be mapped into single butterfly unit. In this paper, an efficient pipelined folded FFT design was implemented for 8 point radix-2 FFT and compared with ordinary 8 point radix-2 FFT design to analyze the efficiency of the algorithm. In the process of decreasing the area of the system, CORDIC algorithm was used for the implementation as it replaces the complex twiddle factors multiplication with shifter and adders [8]. CORDIC algorithm with pipelining and multiplexer unit has been introduced for minimization of the area and critical path delay. In this work, six stage CORDIC algorithm was implemented and examined with several algorithms. From the results, it was concluded that multiplexer based CORDIC with pipelining is very efficient in area as well critical path delay than the CORDIC algorithm without use of the pipelining and multiplexer unit [9]. In CORDIC II algorithm, normal adders were used for implementation and those adder units consume much power as well as area in the FPGA implementation [10]. The above algorithms were still time consuming, hardware requirement and angle set gives less convergence. To conquer these limitations, we proposed an area as well power efficient CORDIC algorithm with less number of adders.

*E-mail address: Sharath.inguva@gmail.com

ISSN: 1791-2377 © 2018 Eastern Macedonia and Thrace Institute of Technology. All rights reserved.

doi:10.25103/jestr.112.14

2. Related Work

Ansuman Mishra et.al[11] has introduced a new sine and cosine function generator by using CORDIC algorithm, which was implemented on ASIC. In DSP applications, there are some set of operations which are frequently used such as addition, multiplication, shifting and trigonometric functions. This algorithm will replace the multiplication process with simple addition, shifting and subtraction operations. Hardware is minimized with absence of flexibility in the hardware implementation.

Prajakta J et.al[12] has proposed a CORDIC algorithm for several digital modulation techniques such as ASK, FSK and PSK. This paper had shown the implementation of digital modulations in MATLAB as well as VHDL by using CORDIC algorithm. For wave form generation Direct Digital Synthesis (DDS) technique was used and it has used the minimum hardware for implementation.

Antonius P Renardyet.al [13] has introduced a conventional CORDIC algorithm with pipelined implementation and Virtually Scaling Free Adaptive (VSFA) CORDIC. In this paper, CORDIC algorithms were implemented on Altera DE2-70 FPGA development board and clearly shown the comparison of various performance parameters such as area and delay. From those comparisons, author has concluded that pipelined CORDIC is best fit for various application over conventional CORDIC algorithm.

Rohit Shukla et.al [14] has introduced a low latency hybrid CORDIC algorithm for reduction of the iterations. With the support of scale factor calculation, rotation and compensation and without huge hardware complexity, number of iterations was minimized. As a result of reduction in the iterations, latency was also minimized.

S. Pongyupinpanich et.al [15] has introduced the CORDIC algorithm for floating point division operations. In the previous algorithms, division operation has a limitation in the range of inputs. So, in this paper, authors have implemented the algorithm for division operation using floating point numbers. With the good convergence, limited input domains of the input ranges was solved for division operations but reconfigurability of core was not possible.

3. Proposed method

The proposed method is the area and power effective solution for any DSP operation implementation on the FPGA. The main aim of the proposed method is to reduce the number of adders and shifting operations in the CORDIC algorithm.

3.1. Rotations in digital systems

Here we will depict some important concepts related to the rotation of the digital systems. In digital systems, rotation by angle β , can be expressed as multiplication by a complex coefficients $P = E+jO$. The block diagram of the CORDIC algorithm is shown in the Fig 1.

$$\begin{bmatrix} Y_D \\ Z_D \end{bmatrix} = \begin{bmatrix} E & -O \\ O & E \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix} \tag{1}$$

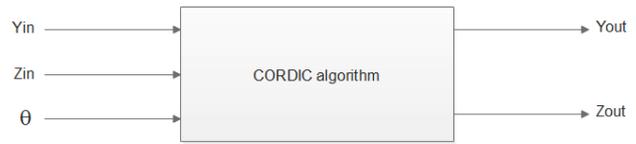


Fig. 1. Block diagram of CORDIC algorithm

where $y + jz$ is the input and $Y_D + jZ_D$ is the output of the rotation. E and O are the n-bit integer numbers in 2's complement within the range $[-2^{n-1}, 2^{n-1} - 1]$, which were obtained by rotation angles as below. Fig 2 shows the rotation of the point on the unit circle.

$$\begin{aligned} E &= A.(\cos \beta + \epsilon_e) \\ O &= A.(\sin \beta + \epsilon_o) \end{aligned} \tag{2}$$

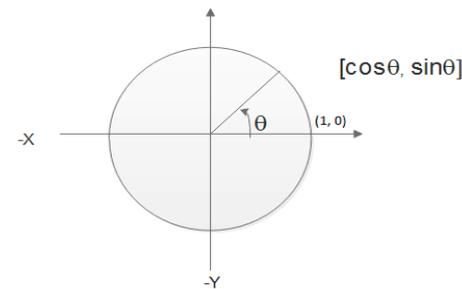


Fig. 2. Rotation of a point in unit circle

Where ϵ_e and ϵ_o are quantization errors of the cosine and sine components and A is the scaling factor. The result Y_D+jZ_D is also scaled by a scale factor A. The rotation error $\epsilon = \sqrt{\epsilon_e^2 + \epsilon_o^2}$ is the distance between the exact rotation and actual rotation due to quantization. If the rotator has multiple rotation angles such as β_i , ($i= 1, \dots, N$), then the respective coefficients $P_i=E_i+jO_i$, for the rotation error will be calculated using the following expression

$$\epsilon = \max(\epsilon(i)) = \max\left(\sqrt{\epsilon_e(i)^2 + \epsilon_o(i)^2}\right) \tag{3}$$

The effective word length is defined as the number of bits of the output result that are guaranteed to be accurate, which is calculated using the rotation error and is expressed below

$$WL = -\log_2 \frac{\epsilon}{2\sqrt{2}} \tag{4}$$

3.2. CORDIC algorithm

The CORDIC algorithm will consider the coefficients $P = E + jO = 2^k + j\delta_k$ where $\delta_k \in \{-1, 1\}$ and $k= 0, 1, \dots, N$ is the micro rotation stages and their respective angles are $\beta_k = \tan^{-1}\left(\frac{O}{E}\right) = \delta_k \tan^{-1}(2^{-k})$.

CORDIC algorithm will split down the rotation angle as below

$$\theta = \sum_{k=0}^N \beta_k + \epsilon_\varphi \tag{5}$$

In the eqn(5) ϵ_k represents the rest of the phase error. Each of the micro rotation stage is calculated by

$$\begin{bmatrix} Y_D \\ Z_D \end{bmatrix} = \begin{bmatrix} 2^k & -\delta_k \\ \delta_k & 2^k \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix} \quad (6)$$

δ_k is the direction of the rotation and scaling factor of the stage is given by $A(k) = \sqrt{2^{2k} + 1}$. At micro rotation stage, rotation error $\epsilon = 0$ and the word length is $WL = \infty$. This describes that the coefficient P_k makes the rotation exactly β_k degrees and scaling factor for both the angles in the each micro rotation will remain same.

4. Angle sets

There are three types of angle sets for the proposed CORDIC algorithm, which are explained in the following:

4.1. Friend angles

There are set of friend angles β_i for which we have a set of coefficients $P_i = E_i + jO_i$, $\beta_i = \tan^{-1}(O_i/E_i)$. The magnitude of P_i is same $\forall i, j, |P_i| = |P_j|$. In this angle set, all the coefficients

have same magnitude. A kernel having friend angles β_i does not have rotation error, for which word length is infinity.

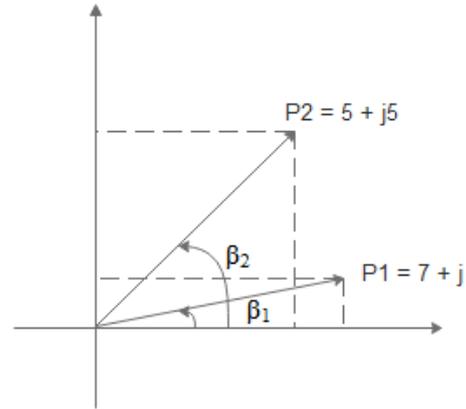


Fig.3. Examples of friend angles

The angles $\beta_1 = 8.13^\circ$ and $\beta_2 = 45^\circ$ are the example of friend angles. For these angles there exist the coefficients $P_1 = 7+j$ and $P_2 = 5+j5$ whose angles are β_1 and β_2 , which are shown in the fig 3 and its magnitudes $|P_1| = |P_2| = \sqrt{50}$. The hardware circuit diagram of the friend angle based CORDIC is shown in the fig 4. The fig 4 does not contain the multiplier and in the implementation it has only shifters, adders and multiplexers.

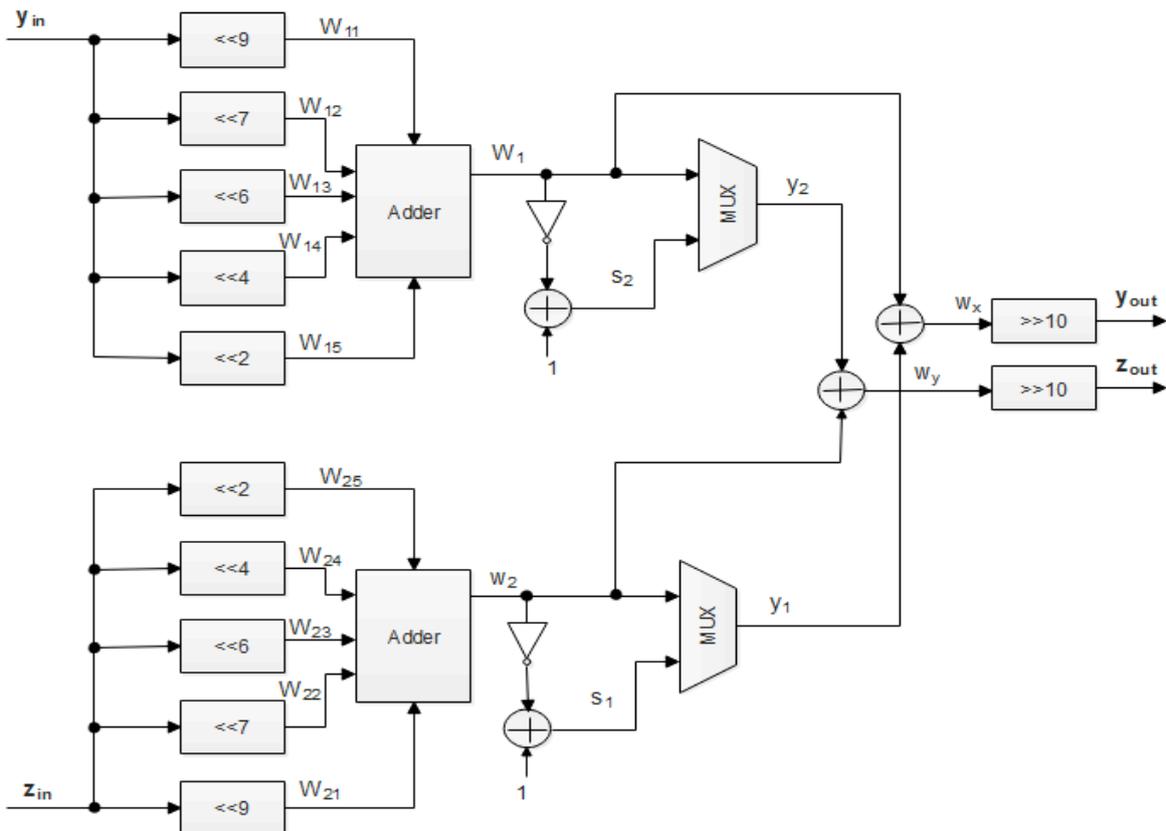


Fig.4. Hardware Circuit of the friend angle based CORDIC

4.2. Uniformly-Scaled Redundant CORDIC

The USR CORDIC rotations will utilize the same rotation angles as the redundant CORDIC with similar scaling of the all the angles. Coefficients of the USR CORDIC are

$$\begin{aligned} P_0 &= 2^{2k-1} + 1 \\ P_1 &= 2^{2k-1} + j2^k \end{aligned} \quad (7)$$

Graphical representation of the USR CORDIC is described in the fig 5. The hardware implementation of the

USR CORDIC rotator is shown in the fig 6. The magnitudes of the coefficients P_0 and P_1 not absolutely same i.e $|P_0| = |P_1| + 1$ and angles of the USR CORDIC are

$$\beta_0 = 0$$

$$\beta_1 = \tan^{-1}\left(\frac{2^k}{2^{2k-1}}\right) = \tan^{-1}\left(2^{-k+1}\right)$$

(8)

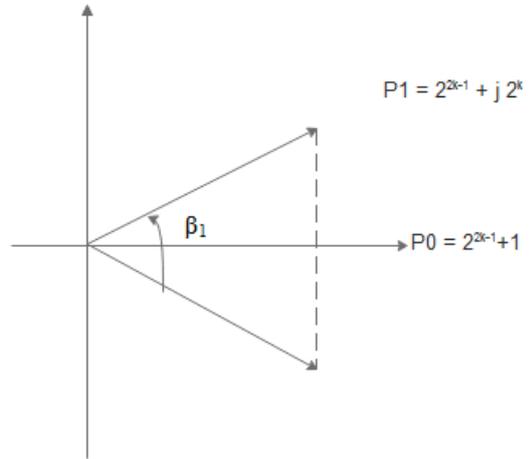


Fig. 5. Graphical representation of USR CORDIC coefficients

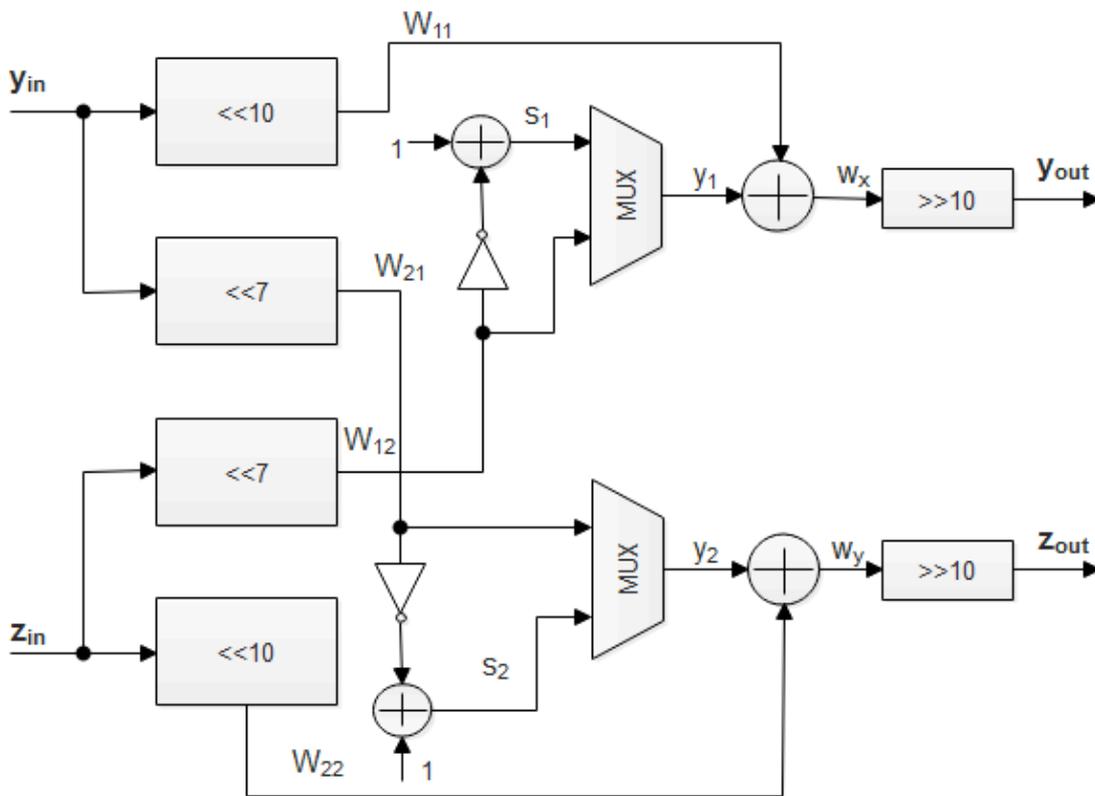


Fig. 6. Hardware Circuit of the USR CORDIC

4.3. Nano rotations

Nano rotations are the kernel which is formed by the coefficient set

$$P_k = E + jk, k = 0, \dots, N \quad (9)$$

Where E is the constant and the respective angles are

$$\beta_k = \tan^{-1}\left(\frac{k}{E}\right) \quad (10)$$

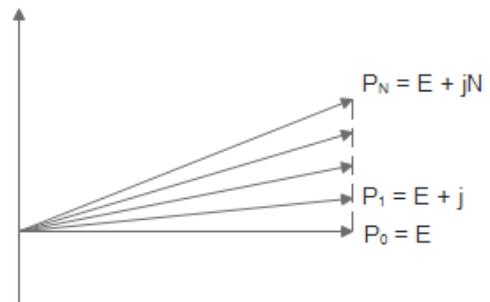


Fig.7. Kernel to calculate nano rotations

N is considered much smaller than E, it will make β_i small and it satisfies the $\beta_i \approx \tan(\beta_i)$. This gives to $\beta_k \approx k/E$, which is a kernel with equally distributed angles. $N \ll E$ makes the scaling of the coefficients very similar. The fig 7 describes the kernel to find nano rotations. We can observe from the fig 7, angle changes by simply changing the value of the imaginary section. With the support of the all above mentioned micro rotations, the proposed method will minimize the count of adders in the implementation. By comparison, In the conventional CORDIC algorithm, the count of adder units will be minimized but we will not be able to reduce the hardware required for the implementation of each adder for various DSP applications. To conquer this,

we have introduced a new Low Area Carry Select Adder(CSLA) to reduce the hardware for implementation of the adder as well as shifters in the proposed method.

The main idea of the proposed system is to use a CSLA adder, instead of using normal adder, which is given in fig.8. This adder can achieve fast arithmetic operation in various data processing technique. CSLA adder is mainly used for reducing area and power dissipation. CSLA is manipulated in many computational structures to cut the carry propagation delay. The elementary knowledge of this work is to have a BEC (binary to excess-1 convertor) instead of RCA (ripple carry adder) with $C_{in}=1$.

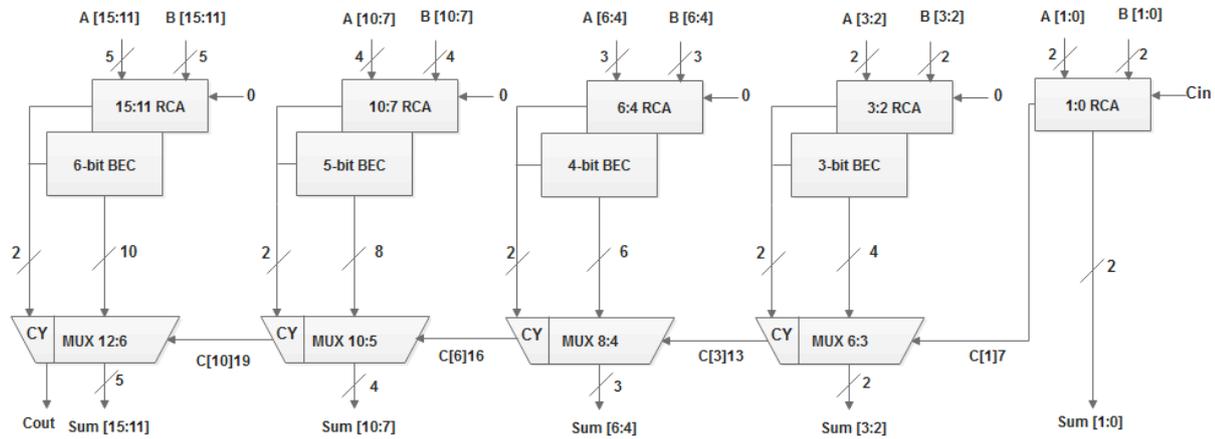


Fig.8. Proposed low area carry select adder

The proposed system using BEC in the place of normal adder or RCA with $C_{in} = 1$ will reduce the power and area. In the place of 2-bit RCA, which has one full adder and one-half adder for $C_{in} = 1$, a 3-bit BEC is utilized which enables to use one bit to the output compared to 2-bit RCA. With this consideration, the time delay has been reduced. It is the process on feedback values, which is the output of the mux is depended on the input of the mux. The input arrival time is lesser than the multiplexer selection input arrival time. By selecting the BEC output or the straight inputs, there are two possibilities is obtainable such as parallel and multiplexer rendering to the regulate signal C_{in} . While designing CSLA, the area will be reduced. The multiplexer delay and mux selection arrival time derived from the different kind of groups. Overall, the power consumption, delay and area will be minimized in the proposed method with the support of CSLA adder.

5.Experimental setup

The proposed method is simulated in Modelsim SE 10.1c using Verilog code. Similarly, determination of area, power, and delay done by using Cadence 180nm technology and RTL compiler, with only shifters and adders instead of a multiplier in the implementation using the CORDIC algorithm.

6.Results and Discussion

The Table 1 depicts the comparison of the existing as well as proposed method by considering the parameters such as

area, power and delay. From this table we can observe that all the parameter, which are mentioned above are minimized. Fig 9 shows the performance comparison graph of the existing as well as proposed system.

Table 1. Performance of area, power and delay for existing and proposed method

CORDIC					
Existing [4]			Proposed		
Area (um ²)	Power(nw)	Delay(ps)	Area(um ²)	Power(nw)	Delay(ps)
134401.2	6489653.4	66445.9	121651.23	5898287.5	42828.5

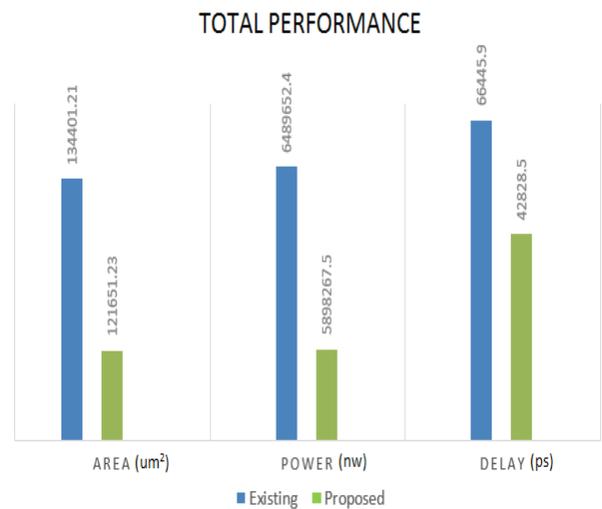


Fig. 9. Performance comparison graph of various parameters

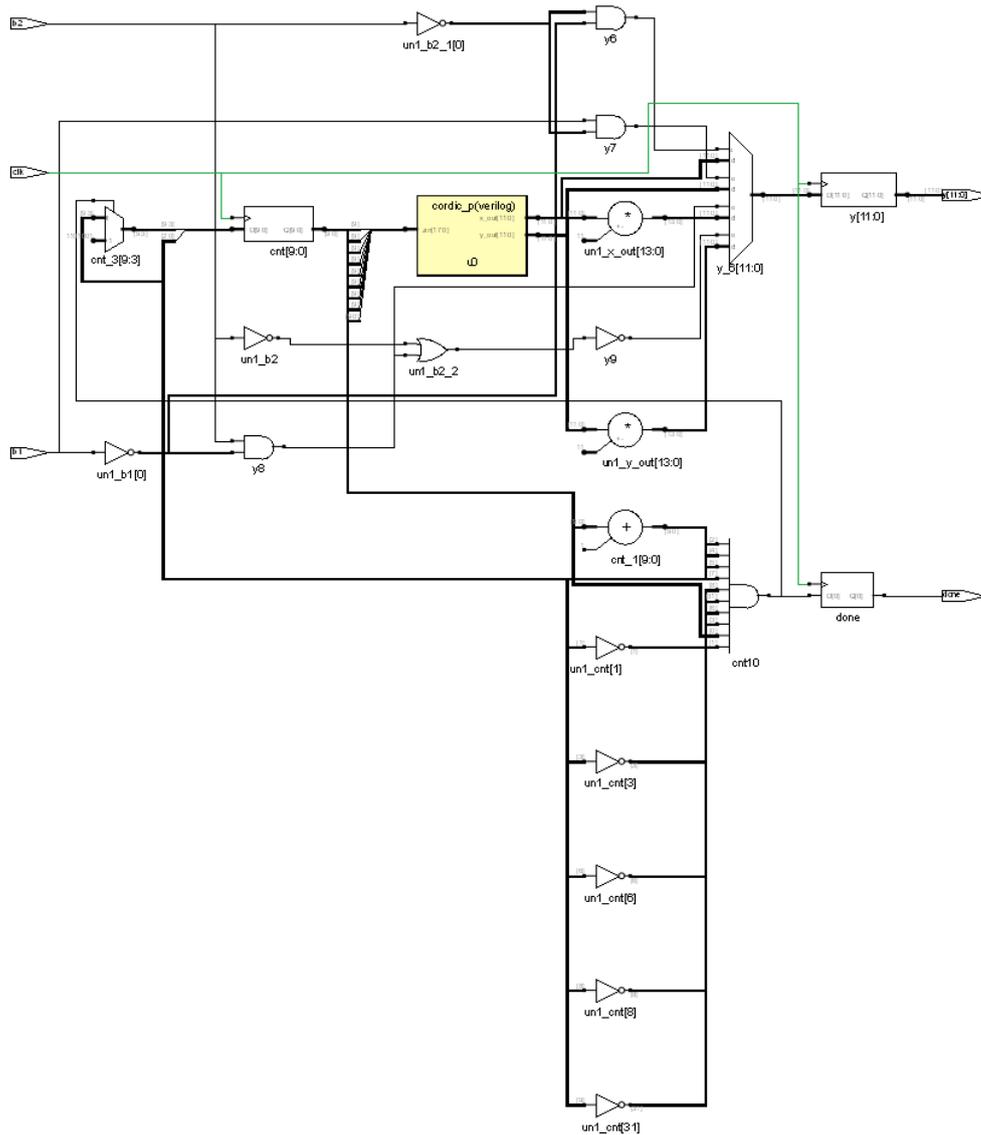


Fig.10. Implementation of QPSK in CORDIC

The implementation of QPSK modulation by using the proposed CORDIC algorithm is shown in fig.10. This diagram is generated by using Synplify pro software and output of the QPSK modulator is shown in the fig 11. The

proposed CORDIC algorithm is used for the implementation of QPSK modulator, to minimize area, power and delay of the QPSK modulators.

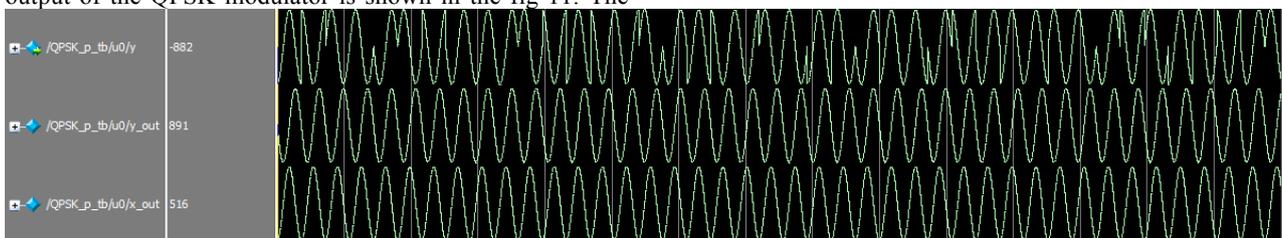


Fig. 11. QPSK modulation output

7. Conclusion

The architecture of the CORDIC is analyzed using different stages of micro rotations, which is used to minimize the angle. The new proposed CORDIC algorithm based upon CSLA adder circuit is used for the implementation of QPSK modulation. By using an efficient CSLA circuit, the

performance parameters such as area, delay and power are minimized in the simulation of proposed CORDIC algorithm.

This is an Open Access article distributed under the terms of the Creative Commons Attribution License



References

- [1] Volder, Jack E., "The CORDIC trigonometric computing technique", IRE Transactions on Electronic Computers, IEEE, pp.330-334,1959.
- [2] Takagi, Naofumi, Tohru Asada, and Shuzo Yajima., "Redundant CORDIC methods with a constant scale factor for sine and cosine computation", IEEE Transactions on Computers, vol.40, no.9, pp.989-995, 1991.
- [3] Moroz, Leonid, TarasMykytiv, and MartynHerasym., "Improved scaling-free CORDIC algorithm", IEEE, Design & Test Symposium, pp. 1-5, 2013.
- [4] Aggarwal, Supriya, Pramod K. Meher, and KavitaKhare., "Concept, Design, and Implementation of Reconfigurable CORDIC", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol.24, no.4, pp.1588-1592, 2016.
- [5] Garrido, Mario, and JesúsGrajal., "Efficient memoryless CORDIC for FFT computation", International Conference on Acoustics, Speech and Signal Processing-ICASSP'07, IEEE, Vol. 2, pp. II-113, 2007.
- [6] Sarode, Namrata, Rajeev Atluri, and P. K. Dakhole., "Mixed-radix and CORDIC algorithm for implementation of FFT", Communications and Signal Processing (ICCSP), 2015 International Conference,IEEE,pp.1628-1634, 2015.
- [7] Parfieniuk, Marek, and Sang Yoon Park., "Sparse-Iteration 4D CORDIC Algorithms for Multiplying Quaternions",vol. 65, no. 9, 2016.
- [8] Krishna, Abhila R., "An efficient folded pipelined architecture for Fast Fourier Transform using Cordic algorithm", Advanced Communication Control and Computing Technologies (ICACCCT), IEEE, International Conference, pp. 462-467, 2014.
- [9] Chinnathambi, M., N. Bharanidharan, and S. Rajaram., "FPGA implementation of fast and area efficient CORDIC algorithm", Communication and Network Technologies (ICCNT), International Conference, IEEE, pp. 228-232, 2014.
- [10] Ibrahim, Muhammad Nasir,Chen Kean Tack,MarianiIdroas,ZuraimiYahya,SitiNoormayaBilmas., "Hardware Implementation of Math Module Based on CORDIC Algorithm Using FPGA", Parallel and Distributed Systems (ICPADS), International Conference,IEEE, pp. 628-632, 2013.
- [11] Mishra, Ansuman, S. Sivanantham, and K. Sivasankaran., "Sine and cosine generator using CORDIC algorithm implemented in ASIC", Online International Conference on Green Engineering and Technologies (IC-GET), IEEE, pp. 1-3, 2015.
- [12] Katkar, Prajakta J., and Yogesh S. Angal., "Realization of cordic algorithm in DDS: Novel Approach towards Digital Modulators in MATLAB and VHDL", Information Processing (ICIP), 2015 International Conference, IEEE, pp. 355-359, 2015.
- [13] Antonius P. Renardy, NurAhmadi, Ashbir A. Fadila, NaufalShidqi, Trio Adiono., "FPGA implementation of CORDIC algorithms for sine and cosine generator", Electrical Engineering and Informatics (ICEEI),International Conference, IEEE, pp. 1-6, 2015.
- [14] Shukla, Rohit, and Kailash Chandra Ray., "Low latency hybrid CORDIC algorithm", IEEE Transactions on Computers, vol.63, no.12, pp.3066-3078, 2014.
- [15] S. Pongyupinpanich, F. A. Samman, and M. Glesner, S. Singhaniyom., "Design and evaluation of a floating-point division operator based on CORDIC algorithm", Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 9th International Conference, IEEE, pp. 1-4, 2012.