International Journal of advancement in electronics and computer engineering (IJAECE)
Volume 6, Issue 12, Dec.2017, pp.1212-1215, ISSN 2278 -1412

Implementation of Efficient CODRIC Architecture Using Carry Skip Adder

Bharti Tiwari[1], Pravin Tiwari[2]
[1]*Mtech. Scholar, Department of ECE, Takshila Institute of engineering and technology, bharti11.tiwari@gmail.com, India;*
[2]*Prof. Department of ECE, Takshila Institute of engineering and technology,pravintiwari@takshshila.org, India;*

Abstract *– Digital Signal process (DSP) applications besides different areas. Additionally a quick, simple, efficient and powerful formula used for varied Digital Signal methodology applications. This temporary present the key idea, approach strategy, and performance architectures which will be organized to work rotation moreover as vectoring-modes. This recommends 3 reconfigurable CORDIC designs. It can, therefore, be wont each circular and hyperbolic CORDIC general reconfigurable CORDIC which will manage in any of the method for every spherical and hyperbolic trajectories. It will use quality over the standard vogue for reconfigurable applications.*

Keywords*: CODRIC architecture, Carry skip adder, Digital Signal process*

I. Introduction

VLSI an extended time, there's nothing new regarding it however as an aspect result of advances within which will be used to style VLSI circuits. The combined result of those 2 advances is that people will currently place numerous functionality into the IC's, gap, wherever intelligent devices are place within everyday objects, and ubiquitous computing wherever little one thing helpful like monitoring your heartbeats. The combined result of those 2 advances is that people will currently place various functionality into the IC's, gap, wherever intelligent devices are put within everyday objects a lot of thought needs to come in actual implementations also as style.

The CORDIC is hardware-efficient algorithms rule for computation of trigonometric and different basic function that uses only modify and increase perform. The CORDIC set of algorithms for the calculation of trigonometric function was intended by Jack E. Volder in 1959 to assist building a time period system for the B-58 bomber. Later, J. Walther in 1971 extended the CORDIC scheme to different functions.

There are 2 modes of CORDIC algorithmic rule, vector translation mode and vector rotation mode. Vector rotations may be used for the conversion of polar to rectangular coordinates. One such algorithmic rule that is extremely a lot of effective for the calculation of trigonometric, hyperbolic, exponential, linear and logarithmic functions is that the CORDIC algorithmic rule. Due to the straightforwardness of concerned operation the CORDIC algorithmic rule is well suitable for VLSI implementation Functions. The algorithmic rule is often derived from the rotation transform.

$x' = x \cos \phi - y \sin \phi$

$y' = y \cos \phi + x \sin \phi$

Rearrangement of above equation can be given as

$x' = \cos \phi [x - y \tan \phi]$

$y' = \cos \phi [y + x \tan \phi]$

The implementation of these equations is still complex trigonometric functions. During the angle conversion phase, the angle is represent as the sum of a decreasing sequence of elementary angles {d = 2s m, i defines a radix number system, 0 = i = -1} where b is the number of rotation, which in turn depends on the accuracy we want, so that In the above algorithmic rule, the parameters constitutes a particular illustration of, and b is the range of bits within the register. At each rotation, the direction of rotation is chosen by obtaining the difference between the actual angle and the angle obtained by rotation.

II. Rotation-Mode Reconfigurable CORDIC

The architecture for reconfigurable rotation-mode CORDIC are shown in figure 1and consists of 3 parts:

1) Preprocessing unit;
2) Reconfigurable CORDIC rotation unit
3) Post processing unit.

The preprocessing unit ensures that the input rotation angle to the CORDIC process structure perpetually lies within the vary [0, π/4], because the most rotation angle which will be handled by micro rotation sequence generator is π/4. The post processing unit is needed just for circular trajectory to swap/complement the sine/cosine values depending on the octant of the rotation angle. The user will control the trajectory of the reconfigurable CORDIC by changing a 1-bit signal T.

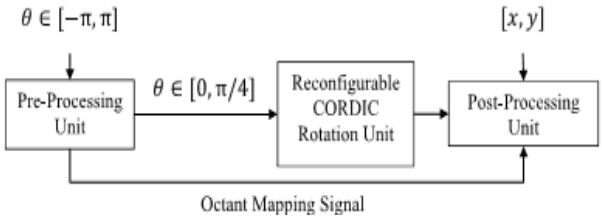

Fig.1 Reconfigurable rotation-mode CORDIC processor

III. Reconfigurable Vectoring-Mode CORDIC

The preprocessing unit ensures that the input rotation angle to the CORDIC process structure perpetually lies within the vary [0, p/4], because the most rotation angle which will be handled by micro rotation sequence generator is p/4. The post processing unit is needed just for circular trajectory to swap/complement the sine/cosine values depending on the octant of the rotation angle, as shown in Figure 2.

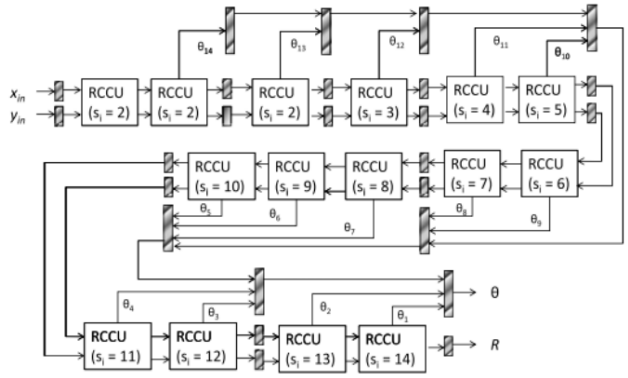

Fig.2 Pipeline reconfigurable vectoring-mode CORDIC unit for s_{basic}=2

IV. Proposed System

The adder of the small rotation sequence generator and reconfigurable coordinate calculation unit is creating the delay due to the conversional methodology. The design needed to totally different adder with low delay. So we tend to tend to choose the carry skip adder for this reason. Since CORDIC is used as a building block in numerous single chip solutions, the critical aspects to be thought of are high speed, low power, and low area, for achieving reasonable overall performance. The design of the projected design of little rotation sequence generator and reconfigurable coordinate calculation unit is given figure 3and 4.

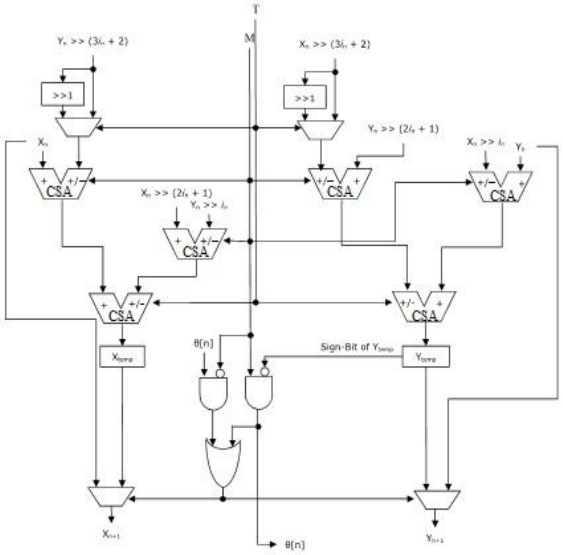

Fig.3 proposed design of micro rotation sequence generator

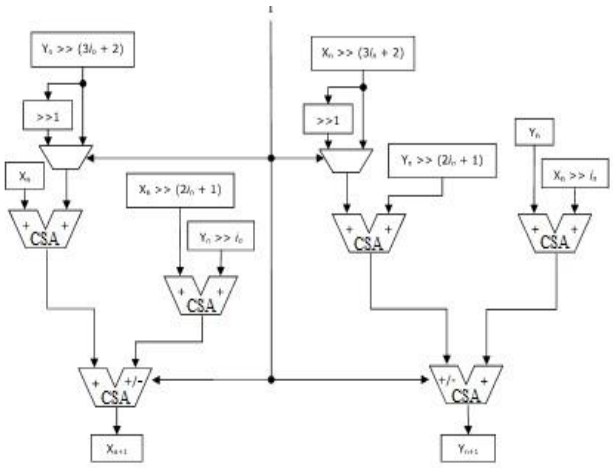

Fig.4 Proposed design reconfigurable coordinate calculation unit

V. Simulation Results

All simulation result and circuitry are presented for the proposed model is in this chapter, in which Cordic Simulation code was synthesized using XILINX and simulation results are shown below:

In figure 5 Top module of recursive module is design in which five input pins are shown and four output pin is shown in which three inputs point contain data and x and y domain input with a reset button for reset circuitry and clock button also for the rotation the x and y input is used besides data input and clock pulse is required for the synchronization.

The above figure 6 is the simulation circuitry RTL view for CORDIC processor using Xilinx ISE simulator. It shows how the output converges to a particular value with the number of iterations.

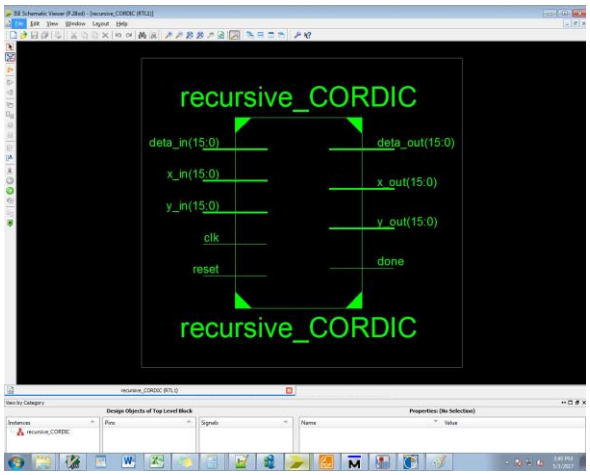

Fig.5 top module

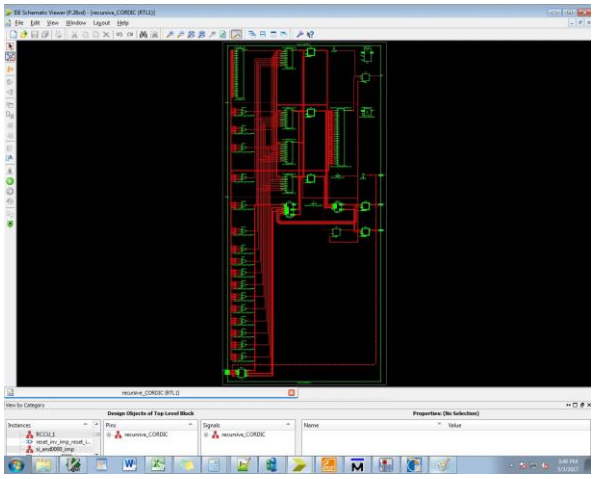

Fig.6 overall design of the CORDIC

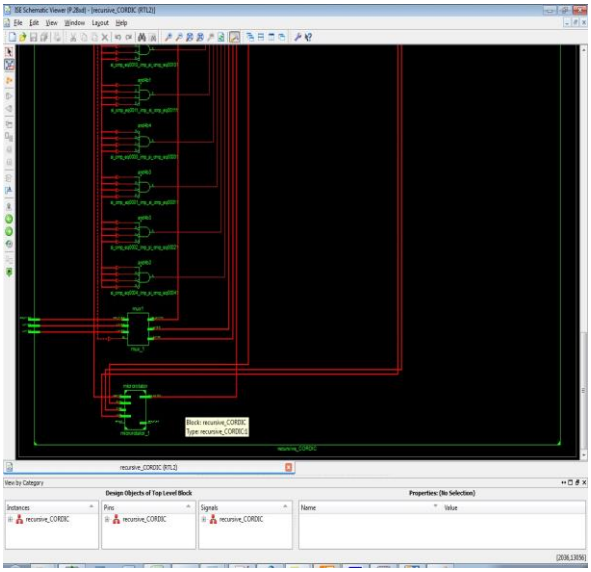

Fig.7 micro rotator unit

In figure 7 Micro rotator unit is shown which is used for the rotation of CORDIC unit function in which three type of rotation is done liner , Circular and hyperbolic.

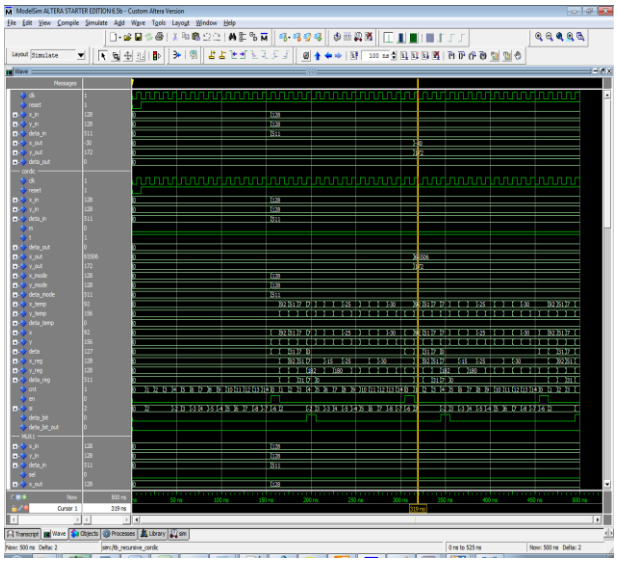

Fig.8 rotation modes circular

Figure 8 the simulation output of proposed system after the circular rotation by micro rotator unit.

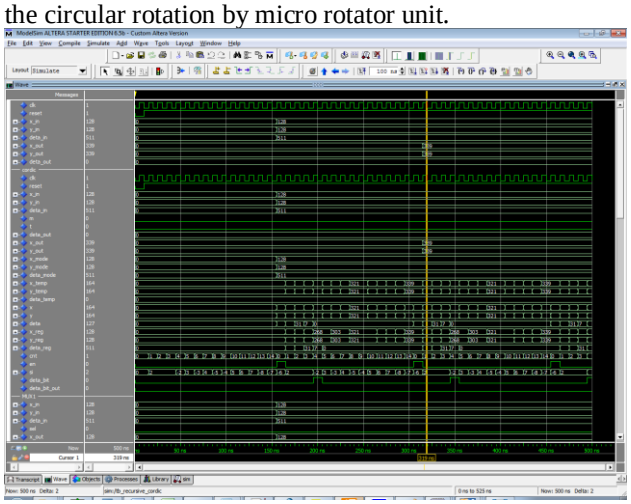

Fig.9 rotation mode: hyperbolic

Figure 9 the simulation output of proposed system after the hyperbolic rotation by micro rotator unit.

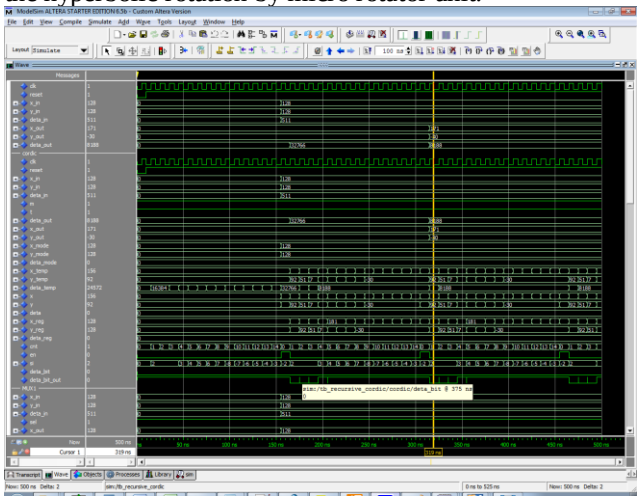

Fig.10 vector mode: circular

Figure 10 gives the vector mode simulation output in circular form of the proposed system.

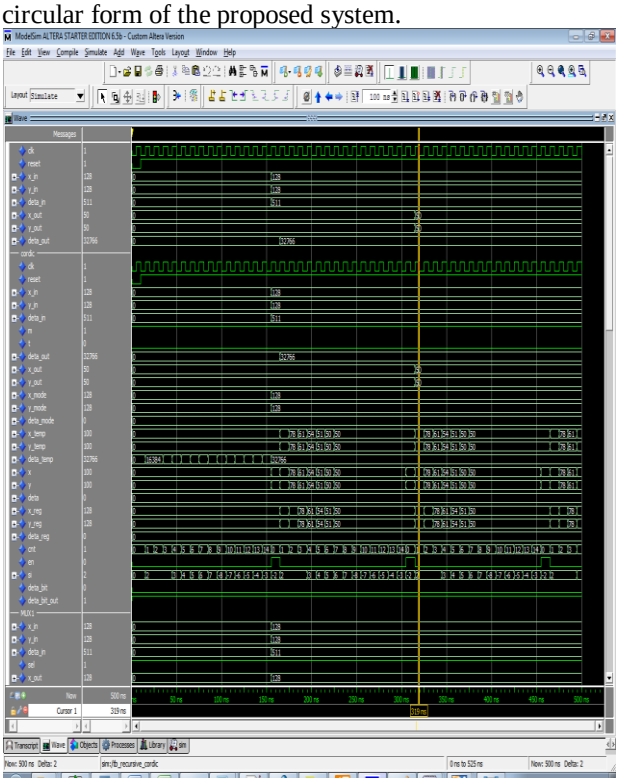

Fig.11 vector mode: hyperbolic

Figure 11 gives the vector mode simulation output in circular form of the proposed system.

Table 1 Comparison table

	Existing design			Proposed design		
	RRMC	RVMC	GRC	RRMC	RVMC	GRC
Slice register	62	66	66	134	134	134
LUT	300	293	409	601	601	601
power consumption (mW)	-	-	-	610	610	610
Delay (ns)	-	-	-	7.566	7.566	7.566

VI. Conclusion

Implementation of a CORDIC-based processor on FPGA provides us a strong mechanism of implementing complicated computations on a platform that gives lots of resources and flexibility at comparatively lesser value. We are reduced delay and space during this projected system. Within the projected system maximum operating frequency are used. The projected reconfigurable CORDICs are utilized in a spread of applications like synchronizers, wave form generators, low value scientific calculators etc.

References

[1] Aggarwal, Supriya, Pramod K. Meher, and Kavita Khare. "Concept, Design, and Implementation of Reconfigurable CORDIC" IEEE Transactions on Very Large Scale Integration (VLSI) Systems 24.4 (2016): 1588-1592.

[2] Vachhani, Leena, K. Sridharan, and Pramod K. Meher. "Efficient CORDIC algorithms and architectures for low area and high throughput implementation" IEEE Transactions on Circuits and Systems II: Express Briefs 56.1 (2009): 61-65.

[3] Kundavaipriya, S., and N. Paruvatham. "Scale-Free Hyperbolic Cordic Processor and Its Application To Waveform Generation" International Journal of Engineering (IJE) Singaporean Journal of Scientific Research (SJSR) 2014.

[4] Chaitanya, K. Naga, and P. Trinatha Rao. "Pipelined Implementation of CORDIC and 64-Point FFT with Memory Interfacing Module"2014.

[5] Meher, Pramod K., et al. "50 years of CORDIC: Algorithms, architectures, and applications." IEEE Transactions on Circuits and Systems I: Regular Papers 56.9 (2009): 1893-1907.

[6] Jain, Rohit Kumar "Design and FPGA Implementation of CORDIC-based 8-point 1D DCT Processor" Diss. Department of Electronics and Communication Engineering, National Institute of Technology, Rourkela, 2011.

[7] Ray Andraka, "FPGA '98" Proceedings of the 1998 ACM/SIGDA sixth international symposium on Field programmable gate arrays, Feb. 22-24, 1998, Monterey, CA. pp191-200.

[8] Vikas Sharma, "FPGA Implementation of EEAS CORDIC based Sine and Cosine Generator", M.Tech Thesis,Dept.Electron.Comm.Eng.,Thapar Uni., Patiala, 2009.

[9] Satyasen Panda,"Performance Analysis and Design of a Discrete Cosine Transform Processor using CORDIC Algorithm", M.Tech Thesis,Dept. Electron.Comm.Eng.,NIT Rourkela, Rourkela, Orissa,2010.

[10] S.K.Pattanaik and K.K.Mahapatra, "DHT Based JPEG Image Compression Using a Novel Energy Quantization Method", IEEE International Conference on Industrial Technology,pp.2827-2832,Dec 2006.

[11] Summanasena M.G.B "A Scale Factor Correction Scheme for CORDIC Algorithm" IEEE, August, 2008.

[12] P. K. Meher, J. Valls, T.-B. Juang, K. Sridharan, and K. Maharatna, "50 years of CORDIC: Algorithms, architectures, and applications," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 56, no. 9, pp. 1893–1907, Sep. 2009.

[13] K. Maharatna, S. Banerjee, E. Grass, M. Krstic, and A. Troya, "Modified virtually scaling-free adaptive CORDIC rotator algorithm and architecture," IEEE Trans. Circuits Syst. Video Technol., vol. 15, no. 11, pp. 1463–1474, Nov. 2005.

[14] F. J. Jaime, M. A. Sánchez, J. Hormigo, J. Villalba, and E. L. Zapata, "Enhanced scaling-free CORDIC," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 57, no. 7, pp. 1654–1662, Jul. 2010 L.

Vachhani, K. Sridharan, and P. K. Meher, "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, Jan. 2009.