International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 3, Issue 6, June 2013)

Scale Free Hyperbolic Cordic Processor using Taylor Series

Subit Abraham[1], Puran Gour[2]
[1,2]Electronics and Communication Department, NRI Institute of Science and Technology Bhopal (M.P.)

***Abstract*— To design a CORDIC processor using hyperbolic functions of sinh and cosh and reduce the scale factor , here we use Taylor series approximation to eliminate scaling in the hyperbolic functions. Here we work in the rotation mode of the algorithm, and a scale free hyperbolic CORDIC processor is designed using VHDL in Xilinx 12.1 ISE Design Suite. Most Significant One bit detection is used for microrotation sequence generation. The input values are converted to 16 bit binary since we are designing for 16 bit Scale free hyperbolic CORDIC Processor. The input coordinates are taken from the LUT initial values .The device used is Virtex 5 XC5VLX330T-2FF1738 in Xilinx to design the CORDIC Processor , 398 slice LUTS are used during the design and number of occupied slices is 257 in the synthesis..**

***Keywords*—Coordinate Rotation Digital Computer (CORDIC), microrotations, rotation mode, scale factor, Taylor series.**

I. INTRODUCTION

Coordinate Rotation Digital Computer is referred to as CORDIC. The main idea of CORDIC arithmetic is based on computational algorithm and it's iterative formulation which was first described by Jack E. Volder in 1959[1], for the computations of trigonometric functions by basic shift and add operations that is calculations are done by shifters and adders.

The CORDIC algorithm operates either in rotation or vectoring mode, following linear, circular or hyperbolic trajectories. The circular CORDIC algorithm is used for computation of sin/cos, vector rotations etc., while hyperbolic CORDIC is used for calculating exponents, sinh/cosh etc. In the rotation mode, the components of a vector and an angle of the rotation are given and the coordinate components of the original vector, after rotation through the given angle, are computed. In the second mode vectoring, the coordinate components of the vector are given and algorithm can be realized as an iterative sequence of additions/ subtractions and shift operations, which are rotations by a fixed rotation angle (sometimes called microrotations) but with variable rotation direction. It's an Hardware Efficient Algorithm. Due to the simplicity of the involved operations the CORDIC algorithm is very well suited for VLSI implementation.

Hyperbolic CORDIC or unified algorithm was first introduced by Walther[2] in 1971 it was an extension to the circular CORDIC by Volder by only changing a few parameters circular CORDIC was extended to a unified algorithm which implemented a wide range of functions including *hyperbolic* , logarithm, exponentials.
For the i^{th} iteration the rotation matrix for the corresponding angle is given by[3]

$$R(i) = K_i \begin{pmatrix} 1 & -\mu_i m 2^{-i} \\ \mu_i 2^{-i} & 1 \end{pmatrix} \quad (0.1)$$

The value of m varies according to the trajectory if circular is used then

m=1 for circular

m=0 for linear

m= -1 for hyperbolic

Here we are considering hyperbolic trajectory.

$$K_i = \frac{1}{\sqrt{1 + m.2^{-2i}}}$$

K_i is the scale factor.

The rotation matrix for the hyperbolic function in the CORDIC algorithm is given as

$$R_p = K_i \begin{pmatrix} 1 & \mu_i 2^{-i} \\ \mu_i 2^{-i} & 1 \end{pmatrix}$$

For the hyperbolic mode of operation (m=-1) for

$\alpha_i = 2^{-i}$

$\alpha_i = \tanh^{-1}(2^{-i})$

The range of convergence (RoC) is not satisfied , the iterations (4, 13,40,121,.....k, 3k+1) should be repeated to attain convergence.

In the scale free CORDIC Algorithm [4] , final target angle is attained by rotating the given vector in one direction only. Therefore the summation of elementary angles is approximated as the final target angle.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 3, Issue 6, June 2013)

With this scale free CORDIC algorithm a foundation is laid to use the approximation of the Taylor series in the CORDIC rotation matrix for the derivations of scale free equations.

The equations for the scale free CORDIC is given as

$$\begin{pmatrix} x_{i+1} \\ y_{i+1} \end{pmatrix} = \begin{pmatrix} 1-2^{-(2i+1)} & -2^{-i} \\ 2^{-i} & 1-2^{-(2i+1)} \end{pmatrix} \begin{pmatrix} x_i \\ y_i \end{pmatrix}$$

Hence the rotation matrix is given as

$$R_p = \begin{pmatrix} 1-2^{-(2i+1)} & -2^{-i} \\ 2^{-i} & 1-2^{-(2i+1)} \end{pmatrix}$$

II. Taylor Series Approximation

The rotation matrix can be written in terms of hyperbolic functions of sinh and cosh as

$$R_p = \begin{pmatrix} \cosh\alpha & \sinh\alpha \\ \sinh\alpha & \cosh\alpha \end{pmatrix}$$

$$\begin{pmatrix} x_{i+1} \\ y_{i+1} \end{pmatrix} = R_p \begin{pmatrix} x_i \\ y_i \end{pmatrix}$$

The Taylor series expansions for sinh and cosh are given as

$$\sinh\alpha_i = \alpha_i + \frac{1}{3!}\alpha_i^3 + \frac{1}{5!}\alpha_i^5 + \frac{1}{7!}\alpha_i^7 + \cdots\cdots\cdots$$

$$\cosh\alpha_i = 1 + \frac{1}{2!}\alpha_i^2 + \frac{1}{4!}\alpha_i^4 + \frac{1}{6!}\alpha_i^6 + \cdots\cdots\cdots$$

For implementations in hardware the above Taylor series expansions of sinh and cosh has to be approximated with a compromise in accuracy at acceptable levels.

Representation of microrotations in hyperbolic mode using Taylor series approximation[5]

$$x_{i+1} = (1+\frac{\alpha_i^{\ 2}}{2!})x_i + (\alpha_i + \frac{\alpha_i^3}{3!})y_i \qquad 1)$$

$$y_{i+1} = (1+\frac{\alpha_i^{\ 2}}{2!})y_i + (\alpha_i + \frac{\alpha_i^3}{3!})x_i$$

$$x_{i+1} = (1+\frac{\alpha_i^{\ 2}}{2!}+\frac{\alpha_i^4}{4!})x_i + (\alpha_i + \frac{\alpha_i^3}{3!}+\frac{\alpha_i^5}{5!})y_i$$

$$y_{i+1} = (1+\frac{\alpha_i^{\ 2}}{2!}+\frac{\alpha_i^4}{4!})y_i + (\alpha_i + \frac{\alpha_i^3}{3!}+\frac{\alpha_i^5}{5!})x_i$$

$$x_{i+1} = (1+\frac{\alpha_i^{\ 2}}{2!}+\frac{\alpha_i^4}{4!}+\frac{\alpha_i^6}{6!})x_i + (\alpha_i + \frac{\alpha_i^3}{3!}+\frac{\alpha_i^5}{5!}+\frac{\alpha_i^7}{7!})y_i$$

$$y_{i+1} = (1+\frac{\alpha_i^{\ 2}}{2!}+\frac{\alpha_i^4}{4!}+\frac{\alpha_i^6}{6!})y_i + (\alpha_i + \frac{\alpha_i^3}{3!}+\frac{\alpha_i^5}{5!}+\frac{\alpha_i^7}{7!})x_i$$

Equation 1) cannot be used in the shift and add operations , to use equation 1) we have to approximate the factorial terms by the power of 2, so we replace 3! by 2^2 in equation 1) we get

$$\begin{pmatrix} x_{i+1} \\ y_{i+1} \end{pmatrix} = \begin{pmatrix} 1+(2!)^{-1}.\alpha_i^{\ 2} & \alpha_i + 2^{-2}.\alpha_i^{\ 3} \\ \alpha_i + 2^{-2}.\alpha_i^{\ 3} & 1+(2!)^{-1}.\alpha_i^{\ 2} \end{pmatrix} \begin{pmatrix} x_i \\ y_i \end{pmatrix}$$

Basic shift determination for a given order of Taylor series approximation

basic shift, $s = \dfrac{b - \log_2(n+1)!}{(n+1)!}$

b is wordlength

n is order of approximation

The basic shift for the third order of approximation using 2) that for n=3 and for wordlength 16 bit is 2.854.

The basic shift cannot be floating point value so the nearest integers are considered as basic shifts so the basic shift will be either 2 or 3.

We have considered basic shift =2 for calculations.

so, $\alpha_{max} = 2^{-s_i}$

$= 2^{-2}$

$=1/4 = 0.25$

in terms of pi : 7pi/88

s_i is number of shifts for i^{th} iteration.Basic shift tells us how many iterations are to be used. The rotation mode gives an angle as well as coordinates for calculation.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 3, Issue 6, June 2013)

III. CORDIC PROCESSOR

We are considering the data in 16 bit and all the values of coordinates and angle are converted to 16 bit binary data. We identify the microrotations depending on the bit representation of the desired rotation angle using most significant one bit detector. For this we restrict the maximum rotation angle to pi/4 radians. As the entire coordinate space [0, 2pi] can be mapped to the [0,pi/4] using octant symmetry of hyperbolic sine and hyperbolic cosine that is sinh and cosh functions. For fixed wordlength of N bit, the shift s_i for the elementary angle is given by

$$s_i = N - MSO_{location} .$$

$MSO_{location}$ is most significant one location. N is wordlength

Rotation angle in terms of n

$$\theta = \sum^{for\_n\_iterations} \alpha_{s_i}$$

$$\alpha_{s_i} = 2^{-s_i}$$

So minimum value of sbasic is 2 , and number of iterations is n =8 ,And the wordlength N= 16 bit , so desired angle of rotation ,

$$\theta = \sum^{n=8} 2^{-2} to 2^{-9} ,$$

Coordinate calculation unit uses the shifts from shift value estimation unit, and employs those values as shifters and then using adders calculates the value the coordinates.

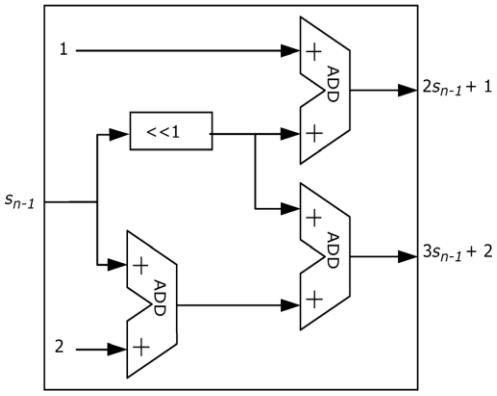

Fig1: Shift Value estimation

In the CORDIC Processor there are three blocks which are as follows:

i. The Micro Rotation Sequence Generator
ii. Generation of Shift Values (Shift Value Estimation)
iii. Coordinate Calculation Unit .

And the microrotation Sequence Generator consists of the Most Significant one(1) bit detector.

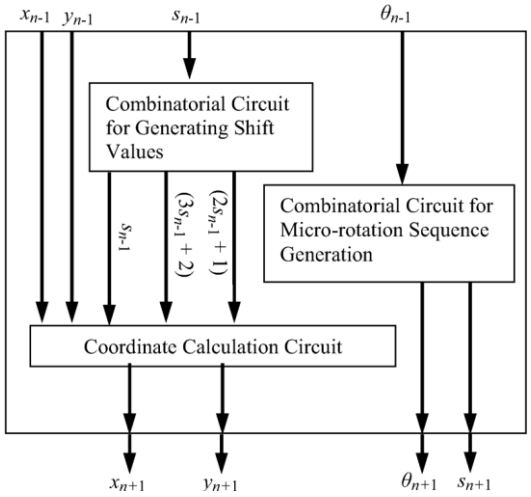

Fig 2 :CORDIC Processor

The coordinate calculation unit calculates the values for the coordinates , given the input coordinate values it calculates the output coordinates x and y . The calculation is done by using shifters and adders. The values for the shifters that how much the value has to be shifted is calculated in the shift value estimation unit . In shift value estimation unit the initial value for the shift is given , and from that value the values for the shifts are calculated . The shift value estimation unit consists of one shifter and adders. Through this shifter and adders the value for the shifts are calculated which are used in the coordinate calculation unit as shifters. Microrotation sequence generator takes the value for input angle and generates the value for the output rotation angle. LUT stores the initial values for the CORDIC Processor, and generates input coordinates. The LUT values which are taken are derived from the octant values. For the Coordinate calculation unit , the LUT values of sinh and cosh are taken, and those values too are converted into 16 bit binary. All these values are given as 16 bit inputs to the CORDIC Processor's components.

IV. CONCLUSION

CORDIC Processor is designed using VHDL Coding. Taylor Series approximation has been used to eliminate scaling in the Hyperbolic Scale free CORDIC Processor.

The device used is Virtex 5 xc5vlx330t-2ff1738 in Xilinx to design the CORDIC Processor , 398 slice LUTS are used during the design and number of occupied slices is 257.

REFERENCES

[1] J. E. Volder, "The CORDIC trigonometric computing technique," IRE Trans. Electron. Comput . , vol. EC-8, pp. 330–334, Sep. 1959.

[2] J. S. Walther, "A Unified algorithm for elementary functions", in Proc. 38th spring Joint Computer Conf., Atlantic City, NJ, 1971,pp 379-385.

[3] P. K. Meher, J. Walls, 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.

[4] 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. 11, no. 11, pp. 1463–1474, Nov. 2005.

[5] S. Aggarwal, P. K. Meher, K. Khare, "Area Time Efficient scaling free CORDIC using generalized microrotation selection," IEEE Trans. Very Large Scale Integration (VLSI) System, vol. 20, no.8, pp 1542-1548, Aug. 2012.