Quest for Research Excellence On Computing, Mathematics and Statistics

Editors
Kor Liew Kee
Kamarul Ariffin Mansor
Asmahani Nayan
Shahida Farhan Zakaria
Zanariah Idrus

Faculty of Computer and Mathematical Sciences

Quest for Research Excellence on Computing, Mathematics and Statistics

Chapters in Book

The 2nd International Conference on Computing, Mathematics and Statistics (iCMS2015)

Editors:

Kor Liew Lee
Kamarul Ariffin Mansor
Asmahani Nayan
Shahida Farhan Zakaria
Zanariah Idrus

Quest for Research Excellence on Computing, Mathematics and Statistics

Chapters in Book

The 2nd International Conference on Computing, Mathematics and Statistics

(iCMS2015)

4-5 November 2015
Langkawi Lagoon Resort
Langkawi Island, Kedah
Malaysia

Publication by
Faculty of Computer & Mathematical Sciences
UiTM Kedah

ISBN 978-967-0314-26-6

Content

CHAPTER 7
Implementation Of CORDIC Algorithm In Vectoring Mode

Anis Shahida Mokhtar, Abdullah bin Mohd Fadzullah

Abstract. Coordinate Rotation Digital Computer (CORDIC) is an algorithm for performing a sequence of iteration computing using the coordinate rotation. The significant of CORDIC lies in the fact that by simple shift-add operations, it can perform computing task such as the calculation of trigonometric, hyperbolic and logarithmic functions, real and complex multiplications, divisions, square roots and many others. In this paper, vectoring mode of CORDIC algorithm was implemented. The algorithm was developed using Verilog HDL in Quartus II software and the results obtained were compared with actual values of the CORDIC algorithm. The highest percentage of difference recorded between the actual value and simulation value was 2.65%.

Keywords: CORDIC, vectoring mode

Anis Shahida Mokhtar (✉) • Abdullah bin Mohd Fadzullah
Department of Electrical & Electronic, Faculty of Engineering
National Defence University of Malaysia
57000 Kuala Lumpur
email:anis@upnm.edu.my, bongzai06@yahoo.com

1 Introduction

In the past few decades, thousands of technologies and inventions, especially in the science and engineering field have been drastically developed. Signal processing and image processing, communications modulations and demodulations, robotics and graphics control, and many other computations based technologies have been developed and improved by researchers, scientists and engineers. Almost of the technologies stated above requires a very high speed calculating algorithm to be implemented with hardware and software implementation. Calculus provides tools and formulas to compute the values of trigonometric functions, for example, series expansions, Taylor polynomial, and rational function approximations [1]. However, these methods require differentiation, integration, multiplication and division operations that make them expensive, space demanding and also much complex when it turns to hardware implementations.

Invention of CORDIC (COrdinate Rotation DIgital Computer) by Jack.E.Volder [1] has been recorded in 1959. His invention was a solution to the navigation computer problems at that time. As time passes, his invention have been studied and developed continuously by other scientists and mathematicians. The beauty of CORDIC is that, by a simple shift-add operations, it is capable to perform several computing tasks such as calculation of trigonometric, hyperbolic and logarithmic functions [2].

2 The CORDIC algorithm

In the most general form, a CORDIC iteration can be written as:

$$x_{i+1} = x_i - m \, . \, \mu_i \, . \, y_i \, . \, \delta_{m,i} \tag{1}$$

$$y_{i+1} = y_i + \mu_i \, . \, x_i \, . \, \delta_{m,i} \tag{2}$$

$$z_{i+1} = z_i - \mu_i \, . \, \alpha_{m,i} \tag{3}$$

Equation (1), (2) and (3) describes a rotation of a plane vector located initially at (x_i, y_i) until (x_{i+1}, y_{i+1}). The third iteration variable z_i keeps track of the rotation angle, $\alpha_{m,i}$. The variable $m \in \{1, 0, -1\}$ specifies a circular, linear or hyperbolic coordinate system, respectively. The rotation direction is steered by the variable $\mu_i \in \{1, -1\}$. There are two operating modes for CORDIC, which is rotation mode and vectoring mode where this paper only focused on the vectoring modes of circular coordinates.

3 Vectoring mode of CORDIC algorithm

Vectoring mode of CORDIC algorithm is to define the angular argument of the initial vector. In vectoring mode of CORDIC operation, the basic idea is to rotate an initial vector (x_i, y_i) and initial z_i=0 until the *y-coordinates* approaches 0 [3].

$$x_{n+1} = x_n - y_n.\sigma_n\,.2^{-n} \tag{4}$$

$$y_{n+1} = y_n + x_n.\sigma_n\,.2^{-n} \tag{5}$$

$$z_{n+1} = z_n - \sigma_n.\tan^{-1}.2^{-n} \tag{6}$$

$$where; \qquad \sigma_n = \begin{cases} 1 \; if \; y_n < 0 \\ -1 \; if \; y_n \geq 0 \end{cases}$$

An initial value of *x* and *y* will be assigned and input *z* will be 0. The input will then iterated using equation (4), (5), and (6) up until *n* times of iterations. *n* will start at 0 until $n - 1$. The output will be the magnitude of x_R, y_R, and z_R, where *K* is the constant gain which is set to *K*=1.64676, as in (7).

$$x_R = K\sqrt{x_i^2 + y_i^2} \tag{7}$$

$$y_R = 0 \tag{8}$$

$$z_R = \tan^{-1}(\frac{y_i}{x_i}) \;, \tag{9}$$

Figure 1 demonstrates an example of rotation trajectory of vectoring mode of CORDIC. In Figure 1, an input vector of v_0 is iterated n^{th} times where *n*=3. The rotation stops at v_3 where it approaches x-axis and the value of angle φ is obtained. As for this CORDIC algorithm, the value of angle φ is the value of angle *z* in radians.

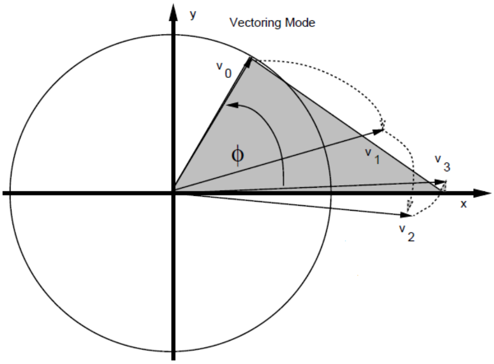

Figure 1: Rotation trajectory of vectoring mode

In this study, the algorithm implemented with some inputs that were randomly assigned. The algorithm also being developed using Verilog HDL in Quartus II software.

4 Methodology

Results produced from the method carried out are actual results calculated using Microsoft Excel and simulation results obtained from the simulation of developed algorithm in Quartus II software and Modelsim-Altera.

In order to perform the calculation and simulation, 20 different inputs were chosen randomly within the first quadrant of Cartesian plane ranging from 0 until 1 as shown in Figure 2.

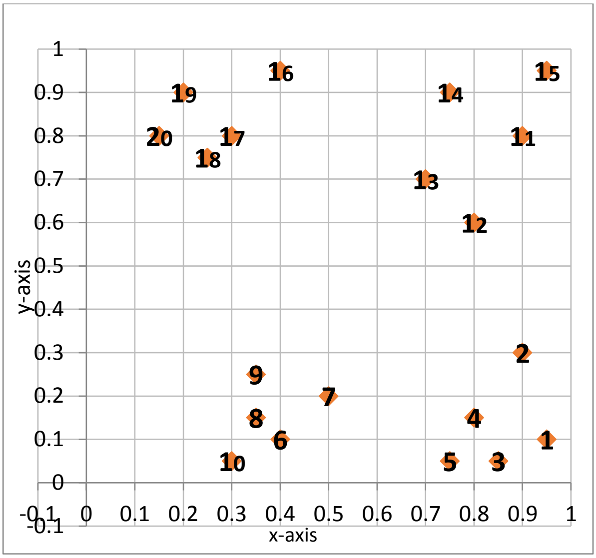

Figure 2: Coordinate of inputs on Cartesian plane

The results obtained from the calculations and the simulations were compared and analyzed in the next section of this paper.

5 Results and Analysis

In this section, results from the method carried out based on the previous section was discussed. 20 different inputs randomly chosen based on Figure 2 were calculated using the vectoring mode of CORDIC algorithm as in Equation (4), (5), and (6). The calculation was made 14 times, from n=0 until n=13. All of the values obtained from the calculation based from inputs assigned were tabulated in Table 1.

The same inputs were used in simulation using Verilog HDL in Quartus II. Simulation was performed based on the developed algorithm of the vectoring mode of CORDIC algorithm. The outputs were verified using Modelsim-Altera. Figure 3 shows the result of the simulation output from Modelsim-Altera.

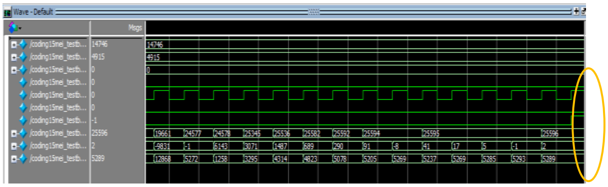

Figure 3: Simulation output of the random input.

From Figure 3 the, the yellow oval indicates that output was read after 14 clock cycles. The values inside the waveform were in the fixed point representation. The simulation was repeated using all the inputs in order to obtain results of all inputs. Note that, the fixed point values will be converted into floating point using *Q1.14* format. The results obtained then recorded and tabulated in Table 1.

Table 3: Table of actual values and simulation obtained from 20 inputs

No	Input			Actual value			Simulation value		
	x	y	z (rad)	x	y	z (rad)	x	y	z (rad)
1	0.95	0.1	0	1.57306553	2.51709E-06	0.10487534	1.57330322	-0.00012207	0.10467529
2	0.9	0.3	0	1.56225395	3.03885E-05	0.3217311	1.56225586	0.00012207	0.32281494
3	0.85	0.05	0	1.40216583	2.33069E-05	0.0587392	1.40216064	0.000244141	0.05718994
4	0.8	0.15	0	1.34036574	-1.3673E-05	0.18535815	1.34033203	6.10352E-05	0.18731689
5	0.75	0.05	0	1.23781175	2.07226E-05	0.06655142	1.237854	0.00012207	0.06488037
6	0.4	0.1	0	0.67897665	6.20603E-06	0.24496952	0.67907715	0.00012207	0.24371338
7	0.5	0.2	0	0.88680754	-1.7875E-05	0.38052653	0.88708496	6.10352E-05	0.37896729
8	0.35	0.15	0	0.62706762	7.89053E-06	0.4048792	0.62713623	6.10352E-05	0.40472412
9	0.35	0.25	0	0.70829837	2.0328E-05	0.62022079	0.70837402	0	0.62237549
10	0.3	0.05	0	0.50084258	-4.541E-06	0.16515774	0.50085449	6.10352E-05	0.16583252
11	0.9	0.8	0	1.98296194	3.21896E-05	0.72662611	1.98309326	6.10352E-05	0.72491455
12	0.8	0.6	0	1.64676026	-2.8975E-05	0.6435187	1.64672852	6.10352E-05	0.64471436
13	0.7	0.7	0	1.63020948	-2.5315E-05	0.78541369	1.63031006	0.000183105	0.78363037
14	0.75	0.9	0	1.92924131	5.79291E-05	0.87602802	1.92944336	6.10352E-05	0.87408447
15	0.95	0.95	0	2.21242716	3.31622E-05	0.78541369	2.2124721	0.000192477	0.78555209
16	0.4	0.95	0	1.69744162	1.55151E-05	1.17226474	1.69750977	0.000244141	1.17095947
17	0.3	0.8	0	1.40699258	6.34519E-05	1.21198056	1.40704346	-6.1035E-05	1.21160889
18	0.25	0.75	0	1.30187829	-2.5324E-05	1.24906522	1.30181885	6.10352E-05	1.25006104
19	0.2	0.9	0	1.51823794	-5.0199E-06	1.35213069	1.51928711	6.10352E-05	1.38348389
20	0.15	0.8	0	1.34036574	9.54826E-05	1.38537714	1.34063721	-6.1035E-05	1.38348389

From Table 1, the percentage of difference between actual value and simulation value were presented using line graph as in Figure 4 until Figure 6.

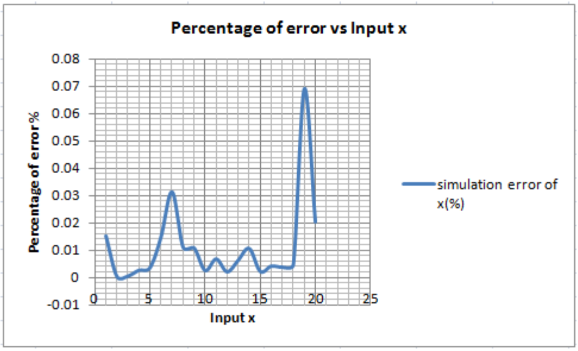

Figure 4 : Percentage of error as a function of Input x

Figure 4 illustrated the percentage of difference between the actual value of x and simulation value of x. The highest point plotted is at x_{19}, where the percentage of error recorded for the value of x is 0.07% error. Referring to Table 1, x_{19} indicates input (0.2, 0.9) for the x and y input. From Figure 3, input (0.2, 0.9) was located farthest from the x-axis. As mentioned earlier in this paper, the same number of 14 iterations and 14 clock cycles were performed for the calculation and simulation respectively. The number of clock cycles sufficient to obtain output with least error was not optimized, thus resulted the 0.075 difference.

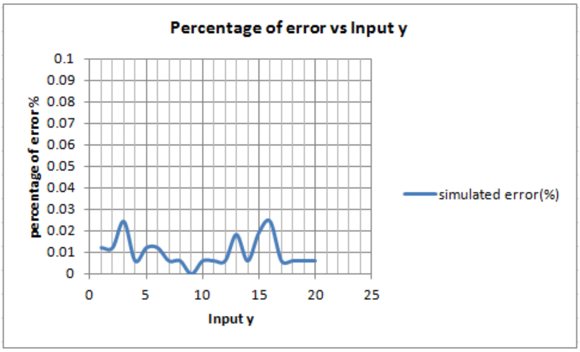

Figure 5: Percentage of error as a function of Input *y*

From Figure 5, it can be seen that the highest percentage of error recorded for the value of y is 0.026% error, at y_{16}. Referring to Figure 2, y_{16} is the input number 16. The number of clock cycles in simulation was insufficient and not optimized.

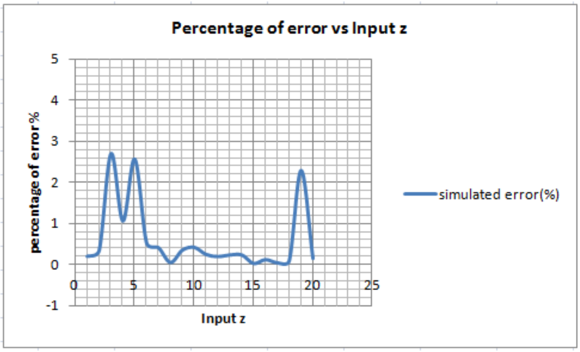

Figure 6: Percentage of error as a function of Input z

As in Figure 6, it illustrates the percentage of difference between the actual value of z and simulation value of z. It can be seen that the highest percentage of error recorded for the value of z is 2.7% error, at z_3. As z was the rotated angle in radian of the input vector, the error was the difference of angle obtained from the actual angle rotated and simulation angle rotated. From Figure 2, input number 3 was the nearest located to the x-axis. As mentioned earlier in this paper, the same number of 14 iterations and 14 clock cycles were performed for the calculation and simulation respectively. The number of clock cycles sufficient to obtain output with least error was not optimized, thus resulted the 2.7% error. It can be optimized, in order to obtain the least difference. However, the optimization between the number of iterations and the clock cycles was not included in the scopes of this study.

After same 14 times iterations for calculation and 14 clock cycles in simulation were performed, the highest difference between the actual value and simulation value recorded is 2.65% which is obtained from the value of z, where z is the value of angle rotated in radians. This error can be considered as low, as many motor control and robotics applications have parameter error tolerance up to ±10% of error [4].

6 Conclusion

The study of the vectoring mode of the CORDIC algorithm were conducted. The results obtained from the calculation using CORDIC algorithm and the simulation using Quartus II software were discussed and analyzed briefly. It is found that the highest error of difference occurred between the actual and simulation value after 14 times of iterations and 14 clock cycles was only 2.65% error. The error occurred because no optimization was made between number of iterations and number of clock cycle. This range of error can be further studied in order to be applied in real time applications either in robotics, signal processing, communication systems, or motor control processor, that are recommended to be conducted in upcoming studies. This is to increase the performance of the design, or the efficiency of the devices, depending on the applications that leaned on this CORDIC algorithm.

References

1. J. E. Volder, "The birth of CORDIC," *J. VLSI Signal Process.,* vol. 25, pp. 101-105, 2000
2. Pramod K. Meher, Javier Valls, Tso-Bing Juang, K. Sridharan and Koushik Maharatna,"50 years of CORDIC: Algorithms, Architecture and Applications," *IEEE Transactions on Circuit and System-I,* vol. 56, no. 9, pp. 1893-1907, Sept. 2009.
3. Herbert Dawid and Heinrich Meyr, "Chapter 24: CORDIC Algorithms and Architectures," retrieved on Nov. 2014, http://www.eecs.berkeley.edu
4. J. E. Volder, "The CORDIC trigonometric computing technique," *IRE Trans. Electron. Computers,* vol. EC-8, pp. 330-334, Sept. 1959.

iCMS 2015

ISBN 978-967-0314-26-6