International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 – 5161
©2015INPRESSCO®, All Rights Reserved
Available at http://inpressco.com/category/ijcet

Research Article

Modified CORDIC Architecture for Fixed Angle Rotation

Binu M Philip†* and Arun P†

†Department of Electronics & Communication Engg., St. Joseph's College of Engineering & Technology, Palai, Kerala, India

Accepted 15 Nov 2015, Available online 20 Nov 2015, **Vol.5, No.6 (Dec 2015)**

Abstract

The fixed and known angle vector rotations have many applications in animation, graphics, games, robotics and digital signal processing. CORDIC requires only simple shift add operation to estimate the basic elementary functions like trigonometric operations, multiplication, division and some other operations like logarithmic functions, square roots and exponential functions. But, there are not many optimized coordinate rotation digital computer (CORDIC) designs for the rotation of vectors through specific angles. Therefore, in this paper, different architectures of CORDIC with different accuracy levels are discussed and present a modified CORDIC architecture. The proposed CORDIC design is synthesized by Xilinx field-programmable gate-array platforms, and shown that the proposed architecture offer higher throughput, less latency and less area-delay product than the reference CORDIC design for fixed and known angles of rotation.

Keywords: *Coordinate rotation digital computer (CORDIC), digital signal processing (DSP) chip, VLSI.*

1. Introduction

CORDIC (for COordinate Rotation Digital Computer) also known as Volder's algorithm was described by Volder in 1959. The key concept of CORDIC arithmetic is based on the basic principles of 2-D geometry (J E Volder, 1959). But the iterative formulation of CORDIC is used for the calculation of trigonometric functions, multiplication, and division.

There are two modes of operation, rotation mode and vectoring mode (B Parhami, 1999). In the first one, rotation mode, coordinate components of vector and rotation angle are given, and from that the coordinate components of original vector, after rotation through the given angle, are computed. In the vectoring mode of operation, the coordinate components of a vector are given and from that the magnitude and the angular argument of the original vector are computed.

The vector rotation angle has many applications in robotics, graphics, games, and animation. Locomotion of robots is very often performed by the rotations through small fixed angles and translations of the links. The translations are performed by simple coordinate value addition while the new coordinates of the next step could be accomplished by suitable successive rotations through a small and fixed angle which could be performed by a CORDIC circuit. There are abundant examples of uniform rotation starting from the basic elements of electronics, i.e, electrons inside an atom to the planets and satellites. An example of uniform rotations is the one degree rotation of the hands of an animated mechanical clock. In some early works, CORDIC circuits have been developed for the implementation of complex multiplications to be used for digital signal processing (DSP) applications (P K Meher, 2009).

Latency of computation is the major issue with the implementation of CORDIC algorithm due to its linear-rate convergence. It requires iterations to have bit precision of the output. Overall latency of computation is directly proportional to the product of the word-length and the CORDIC iteration period. The speed of CORDIC operations is, therefore, constrained either by the iteration count or the duration of the clock period.

The iteration count can be reduced by the application of angle recoding (AR) schemes for CORDIC implementation of constant complex multiplications by encoding the angle of rotation as a linear combination of a set of selected elementary angles of micro-rotations(Y Hu, 1993).

The contribution of this paper is a new and modified architecture that needs less area for implementation and less delay to produce the output.

The remainder of this paper is organized as follows. Section 2 deals with the basic architecture of CORDIC system. Efficient circuits for implementation of micro-rotations for fixed rotations are presented in Section 3. The modified architecture of CORDIC circuit is discussed in Section 4. Section 5 analyzes the synthesis results of the proposed designs and compared with the conventional designs. The conclusion is presented in the next section.

*Corresponding author **Binu M Philip is a PG Scholar and Arun P** is working as Assistant Professor

2. Basic CORDIC System

The basic CORDIC circuit for fixed rotations is shown in Fig. 1. X_0 and Y_0 are fed as set/reset input to the pair of input registers and the successive feedback values X_i and Y_i at i^{th} the iteration are fed in parallel to the input registers. Note that conventionally we feed the pair of input registers with the initial values X_0 and Y_0 as well as the feedback values X_i and Y_i through a pair of multiplexers (P K Meher, 2013).

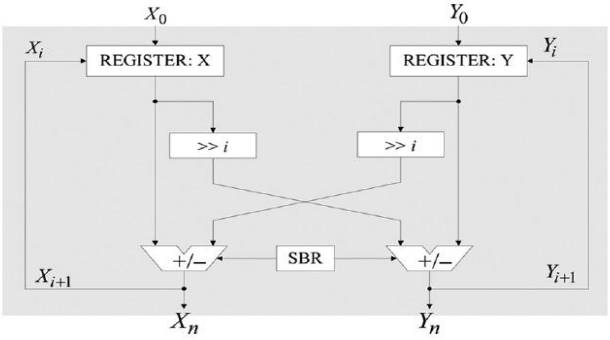


Fig.1 Basic CORDIC circuit for fixed angle rotation

In case of fixed rotation, $\sigma_{i\,i}$ could be pre-computed and the sign-bits corresponding to σ_i could be stored in a sign-bit register (SBR) in CORDIC circuit. The CORDIC circuit therefore need not compute the remaining angle $\emptyset_i$ during the CORDIC iterations. The Fig 1 is directly developed from the equations (1),(2),(3).

$$(U_x)_{i+1} = (U_x)_i - \sigma_i \left(U_y\right)_i 2^{-i} \quad (1)$$

$$(U_x)_{i+1} = (U_x)_i - \sigma_i \left(U_y\right)_i 2^{-i} \quad (2)$$

$$\theta_{i+1} = \theta_i - \sigma_i \tan^{-1}\left(2^{-i}\right) \quad (3)$$

3. Implementation of Micro-Rotations

Since the elementary angles and direction of micro rotations are predetermined for the given angle of rotation, the angle estimation data - path is not required in the CORDIC circuit for fixed and known rotations. Moreover, the corresponding control bits are stored in a ROM because only a few elementary angles are involved. A CORDIC circuit for complex constant multiplications with four micro-rotations is shown in Fig. 2. The ROM contains the control - bits for the number of shifts corresponding to the micro - rotations to be implemented by the barrel - shifter and the directions of micro rotations are stored in the sign-bit register (SBR).

3.1 CORDIC with Four Micro-Rotations

For rotation of a vector through a known and fixed angle of rotation using a rotation-mode CORDIC circuit, we can find a set of a small number of elementary angles $\{\alpha_i$, for $0 \le i \le m\}$, where α_i is the predetermined elementary angle to be used for the i^{th} micro-rotation in the CORDIC algorithm, and m is the minimum number of micro-rotations. It is well known that the rotation through any angle $0 < \theta \le 2\pi$, can be mapped into a positive rotation through $0 < \theta \le \pi/4$ any extra arithmetic operations.

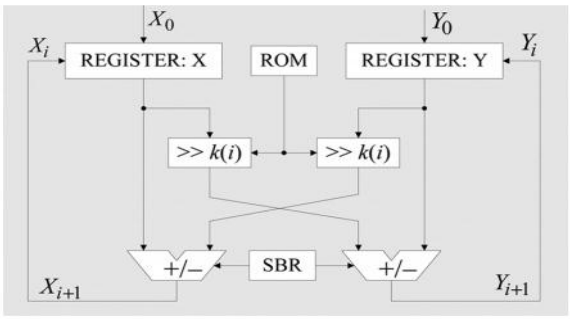


Fig.2 Four micro-rotation CORDIC circuit

Table 1: k(i) and s_i value for different input angles

φ	K(0),s0	K(1),s0	K(2),s0	K(3),s0	ϰφ
45	0,1	-	-	-	0.000
43	0,1	5,0	8,0	-	0.014
41	0,1	4,0	7,0	-	0.024
39	0,1	3,0	6,1	8,1	0.006
37	0,1	3,0	6,0	-	0.020
35	2,1	2,1	2,1	3,0	0.016
33	1,1	3,1	7,0	8,0	0.019
31	1,1	4,1	6,1	-	0.037
29	2,1	2,1	6,1	-	0.032
27	1,1	7,1	-	-	0.013
25	1,1	5,0	8,1	-	0.001
23	1,1	4,0	-	-	0.011
21	2,1	2,1	3,0	10,1	0.003
19	1,1	3,0	7,0	-	0.008
17	1,1	2,0	4,1	6,1	0.000
15	2,1	6,1	10,1	-	0.013
13	1,1	2,0	7,1	-	0.024
11	3,1	4,1	8,1	10,1	0.019
9	3,1	5,1	9,1	-	0.027
7	3,1	9,0	-	-	0.013
5	3,1	5,0	7,0	9,1	0.001
3	4,1	7,0	9,0	-	0.017
1	6,1	9,1	-	-	0.007

S_i is the sign bit, S_i= 1 and 0, for σ_i = 1 and -1

The structure is similar to the basic CORDIC system, but the shift operation is based on the value of k(i) given by the barrel shifter. Here the k(i) can be obtain from table 1. The value of k(i) in each iteration for every angle between 0 to 45 are given here.

The values are equal for two nearby angles, ie, the values of k(i) and s is equal for angle 0 and 1 where sign-bit s is 0 and 1 with respect to the sign of the angle. Then these values are stored in a memory and

given to the shifter as its selection bit. Iteration number is used to shift the vector in Normal CORDIC system.

The Barrel shifter will shift the given vector by a value stored in the ROM as shown in the Fig. 2. After four micro-rotations, the desired sine and cosine value of the given angle can be obtained.

Table 2 K_A valuees for different input angles

Φ	K_A
41	0.7059
39	0.7018
37	0.7018
35	0.9059
33	0.8878
31	0.8926
29	0.9412
27	0.8940
25	0.8940
23	0.8926
21	0.9338
19	0.8878
17	0.8665
15	0.9697
13	0.8682
11	0.9903
9	0.9922
7	0.9922
5	0.9922
3	0.9980
1	0.9999

Each of the input angles requires a starting coordinate at x-axis. Commonly 0.707 is used as the coordinate at x-axis. But it is different for each angle. Table 2 provides K_A value which is used as the starting coordinate for more accuracy and it is also stored in ROM shown in Fig. 2.

3.2 Bi-Rotation CORDIC System

Using only two micro-rotations, it is possible to get accuracy up to 0.033radian. Although the accuracy achieved by two micro-rotations is inadequate in many situations, but can be used for some applications where the outputs are quantized, e.g., in case of speech and image compression, etc.

The bi-rotation CORDIC circuit is shown in Fig. 3. It consists of an adder circuit, two 2:1 multiplexers and a two bit size sign-bit register (SBR). The adder-module contains a pair of adders/subtractors and it performs additions or subtractions according to the value of sign-bit available from the SBR. The components of the input vector are loaded to the input-registers X and Y. The output of the registers are fed to the other modules through two lines where the output is driven to one of the adders/subtractors directly the other line is given to the barrel-shifter, which will shift the input by k(0) bit-locations to right. The outputs of the adders are loaded back to the input registers for the second CORDIC iteration. The bi-rotation CORDIC involves only a pair of barrel-shifters consisting of only one stage of 2:1 MUXes. The control-bit for the barrel-shifters is 0 for the first micro-rotation (no shift) and 1 for the second micro-rotation (shift through k (1) – k (0)).

The values of k (0), k (1) is shown in the table1 and k (1) – k (0) can be calculated using the values from the table1. This values are stored in a memory and given to the shifter which shift the vector by k (0) in 1st iteration and k (1) – k (0) in 2nd iteration.

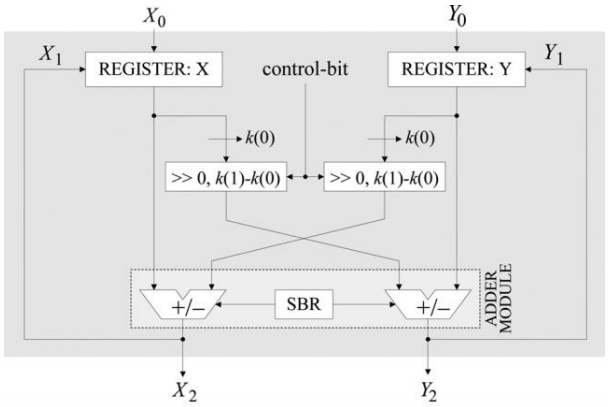
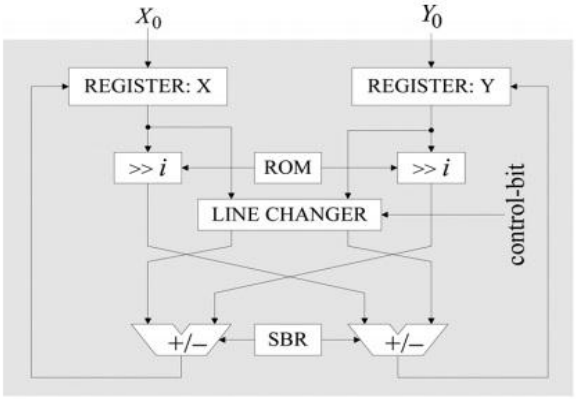
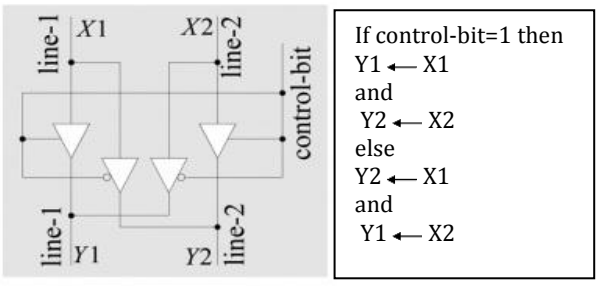


Fig.3 Bi-rotation CORDIC circuit

3.3 CORDIC with Interleaved Architecture



(a)



(b)

Fig.4 CORDIC with interleaved architecture

The approximate scale-factor KA can be estimated according to value of iteration. There are different starting values present for different angles. These values are stored in a memory and used at the start of rotation. This circuit will find the KA values without using a memory. Instead of that, it uses a circuit similar to CORDIC architecture. The Fig.4 can be act as both CORDIC and KA value generating circuit with the use of interleaved architecture.

4. Modified CORDIC Architecture

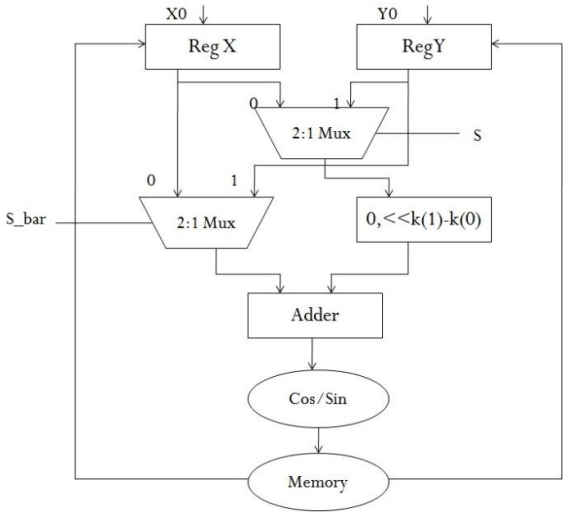


Fig.5 Modified CORDIC architecture

In normal CORDIC structure, the sine section and cosine section are similar in structure, but they use separate components. This increases the delay and usage of FPGA. This can be effectively reduced by the new CORDIC architecture shown in Fig 5. From a single structure, both the sine and cosine values are developed. It uses common adder/subtracter circuit, shifters and memory. The memory will release the values to next iteration only after both the values are stored in the memory.

5. Results and Comparison

5.1 Area optimization

```
HDL Synthesis Report

Macro Statistics
# ROMs                                  : 4
 4x64-bit ROM                          : 4
# Adders/Subtractors                    : 64
 13-bit adder                          : 16
 13-bit adder carry out                : 8
 14-bit adder                          : 24
 15-bit subtractor                     : 16
# Latches                               : 129
 1-bit latch                           : 32
 14-bit latch                          : 1
 4-bit latch                           : 96
# Multiplexers                          : 8
 14-bit 8-to-1 multiplexer             : 8
# Logic shifters                        : 16
 14-bit shifter logical right          : 16
```

Fig.6 Synthesis report of four micro-rotation CORDIC circuit

```
HDL Synthesis Report

Macro Statistics
# ROMs                                  : 4
 4x64-bit ROM                          : 4
# Adders/Subtractors                    : 32
 13-bit adder                          : 8
 13-bit adder carry out                : 4
 14-bit adder                          : 12
 15-bit subtractor                     : 8
# Counters                              : 1
 3-bit up counter                      : 1
# Registers                             : 4
 1-bit register                        : 4
# Latches                               : 137
 1-bit latch                           : 32
 14-bit latch                          : 9
 4-bit latch                           : 96
# Logic shifters                        : 8
 14-bit shifter logical right          : 8
```

Fig.7 Synthesis report of modified CORDIC circuit

The Fig 6 and Fig 7 show the synthesis report of two CORDIC circuits. The modified CORDIC circuit uses only the half of the components of four micro-rotation CORDIC, i.e., adder/subtracter and shifter unit in modified CORDIC is 32 and 8 respectively where as the four micro-rotation circuit uses 64 adder/subtracter units and 16 shifter units.

5.2 Comparison for different circuits

Table 3 Comparison for different circuits

CORDIC	Adders / Subtracters	Shifters/ Multiplexers	Delay(ns)
Basic CORDIC	152	16	8.844
Four micro-rotation	64	24	8.810
Bi-rotation CORDIC	48	18	8.830
CORDIC Interleaved Architecture	128	16	8.926
Modified CORDIC	32	8	7.078

5.3 Simulation result

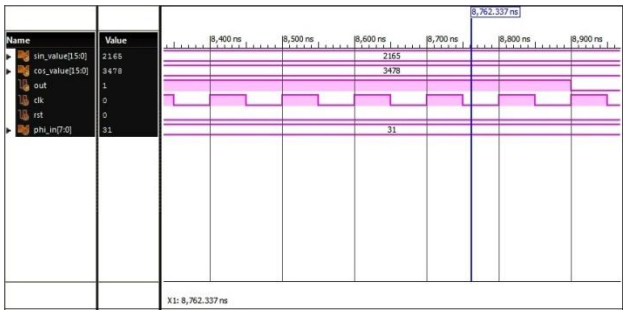


Fig.8 Simulation result of modified CORDIC

Fig 8 shows the simulation result of modified CORDIC where the input angle = 310. The outputs are multiplied by a scaling factor 4096.

Conclusions

Several possible circuits for Rotation mode CORDIC were implemented and an optimized CORDIC architecture is implemented. With this modified structure we are able to reduce the delay and use of components in the FPGA device. The previous delay is 8.810 ns and the modified delay is 7.078 ns. Further this work is extended to design a cascaded CORDIC for their potential for high-throughput and low-latency implementation. We can improve the accuracy in case of small angles of rotation using the same numbers of micro-rotations. The small angle rotators are much useful for shape design and curve tracing for animation and gaming devices.

References

Pramod Kumar Meher, Sang Yoon Park(2013), CORDIC Designs for Fixed Angle of Rotation, *IEEE Transactions on VLSI Systems*, vol. 21, No. 2.

J E Volder (1959),The CORDIC trigonometric computing technique, *IRE Trans.Electron. Comput.*, vol. EC-8, pp. 330-334.

B. Parhami (1999),Computer Arithmetic: Algorithms and Hardware Designs, *2nd edition, oxford*.

Y. H. Hu(1992), CORDIC-based VLSI architectures for digital signal processing, *IEEE Signal Process. Mag*., vol.9, no.3, pp.16-35.

P. K. Meher, J. Valls, T-B Juang, K. Sridharan, and K. Maharatna(2009), 50 Years of CORDIC: Algorithms, Architectures and Applications*, IEEE Transactions on Circuits Systems-I,* vol.56, no.9, pp. 1893-1907.

L. Vachhani, K. Sridharan, and P. K. Meher(2009), Efficient CORDIC algorithms and architectures for low area and high throughput implementation, *IEEE Trans. Circuits Syst. II*, vol. 56, no. 1, pp.61-65.

T. Lang and E. Antelo(2005), High-throughput CORDIC-based geometry operations for 3D computer graphics, *IEEE Trans. Comput.,* vol. 54, no. 3, pp. 347-361.

Y. Hu and S. Naganathan (1993), An Angle Recoding Method for CORDIC Algorithm Implementation, *IEEE Transactions on Computers*, vol. 42, no. 1, pp. 99-102.

H. Li and Y. Xin(2010), Modified CORDIC Algorithm and Its Implementation Based on FPGA, *International Conference on Intelligent Networks and Intelligent Systems*, pp. 618-621,