THE CORDIC ALGORITHM

Fayez Gebali, P.Eng.

May 5, 2014

Contents

1 Introduction

CORDIC is the abbreviation of COordinate Rotation using DIgital Computers. The CORDIC algorithm provides a rich set of elementary functions necessary for scientific and engineering applications [1]. CORDIC-based algorithms are critical to many embedded applications, including motor controls, navigation, signal processing, and wireless communications. The algorithm can be done in software or in hardware. For high-performance applications, the hardware solution is necessary. The main advantage of CORDIC is that it can find many useful functions to a high degree of precision without the need for a multiplier. This lab explores how to implement CORDIC in hardware using VHDL.

We introduce the CORDIC algorithm by first reviewing the equations describing rotating a vector to trace a circle in the plane. This is done in Section 2. We will generalize this discussion by studying rotating a vector to trace a hyperbola in the plane in Section 4

2 Rotating a Vector over a Circle

Rotation implies that the vector traces the circle shown in Fig. 1 and the x and y components of the vector satisfy the equation of the circle:

$$x^2 + y^2 = r^2 \tag{1}$$

where r is the length of the vector which is also the radius of the circle. We can write this vector in Cartesian coordinates as:

$$x_0 = r\cos\theta_0 \qquad y_0 = r\sin\theta_0 \tag{2}$$

Assume that now we want to rotate this vector to the new vector $r\angle\theta_0 + \theta$ through a counterclockwise rotation by an agnle θ. The new values for the x and y components are given by:

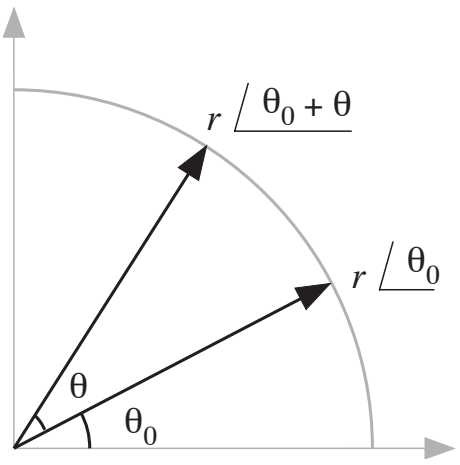

Figure 1: A vector $r\angle\theta_0$ in the $x-y$ plane is rotated counterclockwise by angle θ over a circle of radius r.

$$x_1 = r\cos(\theta_0+\theta) = r\left(\cos\theta_0\cos\theta - \sin\theta_0\sin\theta\right) \tag{3}$$

$$y_1 = r\sin(\theta_0+\theta) = r\left(\sin\theta_0\cos\theta + \cos\theta_0\sin\theta\right) \tag{4}$$

Substituting Eq. (2) in the above two equations we can write:

$$x_1 = x_0 \cos\theta - y_0 \sin\theta \quad (5)$$
$$y_1 = y_0 \cos\theta + x_0 \sin\theta \quad (6)$$

A computer that is required to perform this vector rotation needs to perform four multiplication operations. The problem here is that multiplication operation is expensive in area, delay and energy consumption and several techniques were developed to circumvent this. One approach used by CORDIC is to first factor out the cosine term $\cos\theta$:

$$x_1 = \cos\theta\,(x_0 - y_0 \tan\theta) \quad (7)$$
$$y_1 = \cos\theta\,(y_0 + x_0 \tan\theta) \quad (8)$$

We can even go one step further if we deal only with the scaled values $x_1/\cos\theta$ and $y_1/\cos\theta$ and perform the multiplication by $\cos\theta$ later:

$$x_1 = x_0 - y_0 \tan\theta \qquad y_1 = y_0 + x_0 \tan\theta \quad (9)$$

Now the rotation operation involves only two multiplications.

CORDIC introduces two clever points to perform the simple *scaled* rotations in Eq. (9):

1. The angle θ is decomposed using a set of special rotation angles:

$$\{\ \theta_0 \quad \theta_1 \quad \cdots \quad \theta_{n-1}\ \} \quad (10)$$

   The maximum range of the angles that can be represented by this set of special angles is given by:

$$\theta_{max} = \theta_0 + \theta_1 + \cdots + \theta_{n-1} \quad (11)$$
$$\theta_{min} = -\theta_0 - \theta_1 - \cdots - \theta_{n-1} \quad (12)$$

   Using this set of angles, we can express any angle $\theta_{min} \le \theta \le \theta_{max}$ as:

$$\theta = \ \mu_0\theta_0 + \quad \mu_1\theta_1 + \quad \cdots + \quad \mu_{n-1}\theta_{n-1} \quad (13)$$

   where the $\mu_i = \pm 1$ is chosen to satisfy the above equation.

2. The special angles θ_i are chosen so that the multiplications in (9) are simple right-shift operations!

$$\tan\theta_i = 2^{-i} = \delta_i, \qquad i = 0, 1, \cdots n-1 \quad (14)$$
$$\theta_i = \tan^{-1} 2^{-1} = \tan^{-1}\delta_i \quad (15)$$

   These special angles are pre-calculated and stored in a lookup table.

So, if we want to rotate our vector by angle θ_i, we only need to do the shift/add operations over a set of iterations:

$$x_{i+1} = x_i - y_i\, 2^{-i} = x_i - \mu_i\, y_i\, \delta_i \quad (16)$$
$$y_{i+1} = y_i + x_i\, 2^{-i} = y_i + \mu_i\, x_i\, \delta_i \quad (17)$$

where $i = 0, 1, \cdots, n-1$ and μ_i indicates the direction of rotation: $\mu_i = +1$ for counterclockwise rotation at iteration i and $\mu_i = -1$ for clockwise rotation at iteration i. To keep track of the rotation angle we need to perform, we introduce the new variable z that stores the rotation angle:

$$z_{i+1} = z_i - \mu_i \, \theta_i \tag{18}$$

After rotating counter clockwise by an angle θ_i, the *remaining* rotation angle is given by z_{i+1}. We would know that we completed our rotations when $z_{i+1} \to 0$.

The true value for x_{i+1} or y_{i+1} at the end of iteration i is easily obtained by the formulas:

$$x_{i+1}(\text{true}) = \cos\theta_i \; x_{i+1} \tag{19}$$

$$y_{i+1}(\text{true}) = \cos\theta_i \; y_{i+1} \tag{20}$$

Notice that this scale factor $k_i = \cos\theta_i$ is independent of the direction of rotation. Furthermore, we can defer multiplying by the scale factor till the end of all the iterations. This is discussed in Section 5.2.

2.1 Basic Circular CORDIC Iterations

The basic circular CORDIC equations at each iteration are now summarized as:

$$x_{i+1} = x_i - \mu_i \, y_i \, \delta_i \tag{21}$$

$$y_{i+1} = y_i + \mu_i \, x_i \, \delta_i \tag{22}$$

$$z_{i+1} = z_i - \mu_i \, \theta_i \tag{23}$$

where μ_i indicates direction of rotation (+1 for counter clockwise and -1 for counter clockwise rotation), $\delta_i = 2^{-i}$ and $\theta_i = \tan^{-1}\delta_i$. Table 1 shows the angles θ_i corresponding to $\tan\theta_i = 2^{-i}$ with $i = 0, 1, \cdots n-1$. Note that the value of 2^{-i} and θ_i in radians quickly become almost equal.

3 Rotating a Vector over a Line

Figure 2(a) shows a hyperbola where any point (x, y) on it satisfies the equation:

$$x = r \tag{24}$$

Here r is the distance of the vertical line from the y-axis. Note that Eq. (24) shows similarities to (1) except the y entry is missing.

A point (x_0, y_0) on the straight line is described by the equations:

$$x_0 = r \qquad y_0 = r\delta_0 \tag{25}$$

where the angle now is defined as $\theta_0 = y_0/x_0 = y_0/r$ and $\theta_0 = \delta_0$.

Rotating this point counterclockwise by an angle θ results in the new point (x_1, y_1) with larger values for y_1:

$$x_1 = r \tag{26}$$

$$y_1 = r(\theta_0 + \theta) = y_0 + r\delta \tag{27}$$

Table 1: Values of θ_i corresponding to $\tan\theta_i = 2^{-i}$ with $i = 0, 1, \cdots 15$.

i	$\delta_i = 2^{-i}$	θ_i (radians)	θ_i (degrees)
0	1	0.79	45
1	0.5	0.46	27
2	0.25	0.24	14
3	0.12	0.12	7.1
4	0.062	0.062	3.6
5	0.031	0.031	1.8
6	0.016	0.016	0.9
7	0.0078	0.0078	0.45
8	0.0039	0.0039	0.22
9	0.002	0.002	0.11
10	0.00098	0.00098	0.056
11	0.00049	0.00049	0.028
12	0.00024	0.00024	0.014
13	0.00012	0.00012	0.007
14	6.1e-05	6.1e-05	0.0035
15	3.1e-05	3.1e-05	0.0017

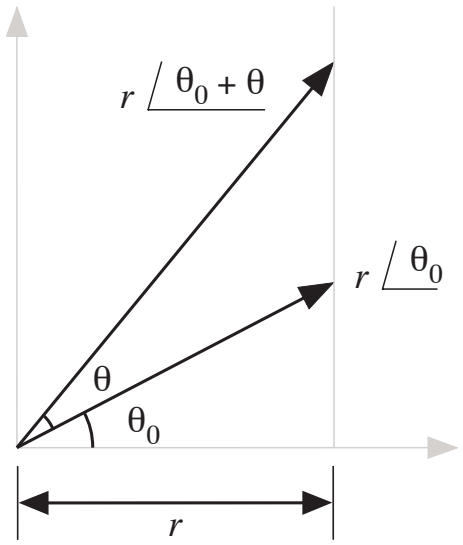

Figure 2: A vector in the $x - y$ plane rotating over a straight vertical line.

We can also define the special linear rotation angles θ_i that satisfy the following equations:

$$\theta_i = 2^{-i} = \delta_i, \qquad i = 0, \cdots n-1 \tag{28}$$

We can change the direction of rotation at iteration i using the variable μ_i to get the CORDIC iterations in linear mode as:

$$x_{i+1} = x_i \tag{29}$$
$$y_{i+1} = y_i + \mu_i \, x_i \, \delta_i \tag{30}$$
$$z_{i+1} = z_i - \mu_i \, \theta_i \tag{31}$$

where $\mu_i = \pm 1$ and $\theta_i = \delta_i$ with $i = 0, 2, \cdots, n-1$. A counter clockwise rotation implies $\mu_i = +1$ and the value of y increases and the value of the remaining rotation angle decreases.

4 Rotating a Vector over a Hyperbola

Figure 3(a) shows a hyperbola where any point (x, y) on it satisfies the equation:

$$x^2 - y^2 = r^2 \tag{32}$$

Here r is the distance between the vertex and the origin of coordinates, as shown in the figure. Note that Eq. (32) shows similarities to (1) except for the negative sign. The x and y coordinates of the point on the hyperbola in Fig. 3(a) can be described in terms of r using equations very similar to Eq. (2).

$$x = r \cosh\theta \qquad y = r \sinh\theta \tag{33}$$

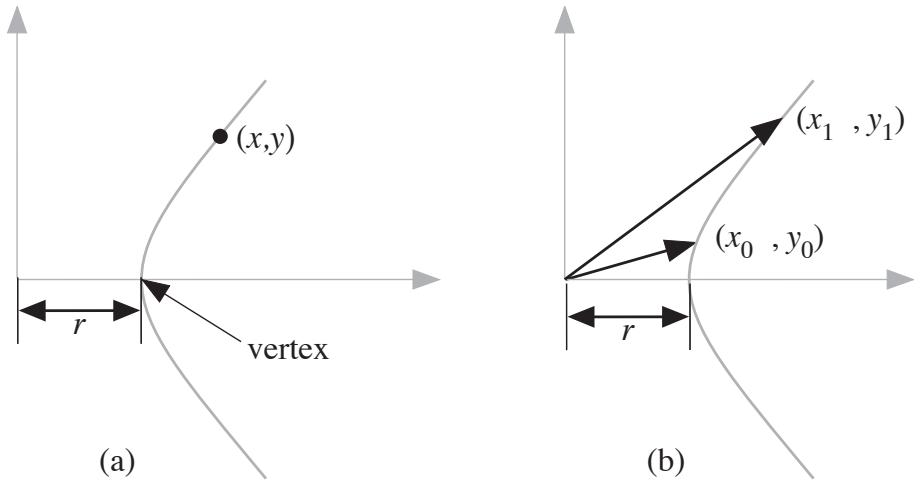

Figure 3: A vector in the $x - y$ plane rotating over a hyperbola. (a) Illustrating the hyperbola "radius" r. (b) Illustrating "rotating" a vector along the hyperbola.

The "angle" θ here could now be defined using the ratio of the x and y coordinates:

$$\tanh\theta = \frac{y}{x} = \frac{\sinh\theta}{\cosh\theta} \tag{34}$$

When the angle $\theta = 0$, our vector lies on the hyperbola vertex. When the vector is rotated over the hyperbola counterclockwise by an "angle" θ, the vector is scaled by the factors $\cosh\theta$ and $\sinh\theta$, for x and y coordinates, respectively. Conversely, clockwise rotation of the vector traces the lower portion of the hyperbola.

Figure 3(b) shows the point (x_0, y_0) described by the equations:

$$x_0 = r\cosh\theta_0 \qquad y_0 = r\sinh\theta_0 \tag{35}$$

Rotating this point counterclockwise by an angle θ results in the new point (x_1, y_1) with larger values for both x_1 and y_1:

$$x_1 = r\cosh(\theta_0 + \theta) = r\left(\cosh\theta_0\cosh\theta + \sinh\theta_0\sinh\theta\right) \tag{36}$$
$$y_1 = r\sinh(\theta_0 + \theta) = r\left(\sinh\theta_0\cosh\theta + \cosh\theta_0\sinh\theta\right) \tag{37}$$

Substituting Eq. (35) in the above two equations we can write:

$$x_1 = x_0\cosh\theta + y_0\sinh\theta \tag{38}$$
$$y_1 = y_0\cosh\theta + x_0\sinh\theta \tag{39}$$

Similar to circular rotations, we can now express our simplified hyperbolic rotations as:

$$x_1 = x_0 + y_0\tanh\theta \tag{40}$$
$$y_1 = y_0 + x_0\tanh\theta \tag{41}$$

We can also define the special hyperbolic rotation angles θ_i that satisfy the following equations:

$$\tanh\theta_i = 2^{-i} = \delta_i, \qquad i = 1, \cdots n-1 \tag{42}$$

$$\theta_i = \tanh^{-1}\left(2^{-i}\right) \tag{43}$$

We know that $-1 < \tanh\theta < 1$. Therefore, $i = 0$ is not allowed since $\tanh^{-1}(1) = \infty$.

So, if we want to rotate our vector by angle θ_i, we only need to do the shift/add operations over a set of iterations:

$$x_{i+1} = x_i + \mu_i\, y_i\, 2^{-i} \tag{44}$$

$$y_{i+1} = y_i + \mu_i\, x_i\, 2^{-i} \tag{45}$$

where $\mu_i = \pm 1$ with $i = 1, 2, \cdots, n-1$. To keep track of the net rotation angle we actually performed, we introduce the new variable z that stores the net rotation angle:

$$z_{i+1} = z_i - \mu_i\, \theta_i \tag{46}$$

4.1 Basic Hyperbolic CORDIC Iterations

The basic hyperbolic CORDIC equations at each iteration are now summarized as:

$$x_{i+1} = x_i + \mu_i\, y_i\, \delta_i \tag{47}$$

$$y_{i+1} = y_i + \mu_i\, x_i\, \delta_i \tag{48}$$

$$z_{i+1} = z_i - \mu_i\theta_i \tag{49}$$

where $\mu_i = \pm 1$ with $i = 1, 2, \cdots\ n-1$, $\delta_i = 2^{-i}$ and $\theta_i = \tanh^{-1}\delta_i$.

Table 2 shows the angles θ_i corresponding to $\tanh\theta_i = 2^{-i}$ with $i = 1, \cdots n-1$. Note that the value of 2^{-i} and θ_i in radians quickly become almost equal.

5 The Generalized CORDIC algorithm

CORDIC is an iterative algorithm that modifies three sets of numbers x, y and z at each iteration. The number of iterations n equals the number of bits of the data (assuming fixed-point data). The CORDIC equations are general in the sense that we can describe circular as well as hyperbolic rotations through the *mode* variable m. We can write the curve over which a point moves as:

$$x^2 + my^2 = r^2 \tag{50}$$

When $m = 1$, have circular rotation as discussed in Section 2. When $m = -1$, we have hyperbolic rotation as discussed in Section 4. CORDIC goes even a step further by defining a *linear* rotation when $m = 0$ where the point now moves along a vertical line defined by $x = r$.

The generalized CORDIC iterations can be summarized as follows

$$x_{i+1} = x_i - m\,\mu_i\, y_i\, \delta_i \tag{51}$$

$$y_{i+1} = y_i + \quad \mu_i\, x_i\, \delta_i \tag{52}$$

$$z_{i+1} = z_i - \quad \mu_i\, \theta_i \tag{53}$$

Table 2: Values of θ_i corresponding to $\tanh \theta_i = 2^{-i}$ with $i = 1, \cdots 15$.

i	$\delta_i = 2^{-i}$	θ_i
1	0.5	0.55
2	0.25	0.26
3	0.12	0.13
4	0.062	0.063
5	0.031	0.031
6	0.016	0.016
7	0.0078	0.0078
8	0.0039	0.0039
9	0.002	0.002
10	0.00098	0.00098
11	0.00049	0.00049
12	0.00024	0.00024
13	0.00012	0.00012
14	6.1e-05	6.1e-05
15	3.1e-05	3.1e-05

where the index of iteration depends on the operation being performed as explained by Eq. (57). The two variables x and y represent the components of the vector in the plane. The new variable z corresponds to the rotation angle whether this rotation angle is in circular, linear, or hyperbolic modes. In the above equations m is the mode and and δ_i, θ_i and the iteration index i are given by:

$$m = \begin{cases} 1 & \text{circular mode} \\ 0 & \text{linear mode} \\ -1 & \text{hyperbolic mode} \end{cases} \tag{54}$$

$$\delta_i = 2^{-i} \tag{55}$$

$$\theta_i = \begin{cases} \tan^{-1}\delta_i & m = 1 \\ \delta_i & m = 0 \\ \tanh^{-1}\delta_i & m = -1 \end{cases} \tag{56}$$

$$i = \begin{cases} 0,\ 1,\ \cdots\ n-1 & m = 1 \\ 0,\ 1,\ \cdots\ n-1 & m = 0 \\ 1,\ 2,\ \cdots\ n-1 & m = -1 \end{cases} \tag{57}$$

The magnitude of θ_i values for $m = \pm 1$ are precomputed and stored in lookup tables.

For the case of $m = -1$ (hyperbolic mode), some iterations have to be repeated twice at values of i given by:

$$i = 4,\ 13,\ ,40,\ \cdots,\ \left(3^{j+1} - 1\right)/2,\ \cdots \qquad j = 1,\ 2,\ 3, \cdots \tag{58}$$

for the case when $n = 32$, the hyperbolic repeated iterations are needed at iterations:

$$i = \ 4, \quad 13 \tag{59}$$

CORDIC algorithm requires the following operations in each iteration:

- One table lookup access
- Two shifts
- Three additions

The variables x and y represent the coordinates of a vector in two-dimensional space. The variable z accumulates the rotated angle of the vector such that when the vector rotates counterclockwise, θ is positive and vice versa. This convention allows us to extract the angle the vector makes with the x-axis as a positive value. It also allows us to rotate the vector anticlockwise with a given angle.

CORDIC has three *modes*, determined by the choice of m, and two *operations* depending whether we want to iteration until variable y is zeroed or variable z is zeroed. In the vectoring mode, the goal is to make the value of y_i go to zero at each iteration. In the rotation mode, the goal is to make the value of z_i go to zero at each iteration. Table 3 summarize the CORDIC modes and operations.

The direction of rotation μ at each iteration is determined the operation being performed as Table 4 shows.

Table 3: CORDIC modes and possible operations.

	Comment
Circular mode	$m = 1$
Linear mode	$m = 0$
Hyperbolic mode	$m = -1$
Vectoring operation	$y \to 0$
Rotation operation	$z \to 0$

Table 4: Guidelines for choosing the direction of rotation.

Vectoring Operation ($y \to 0$)	Rotation Operation ($z \to 0$)
When $y_i < 0 \to$ counterclockwise rotation ($\theta > 0$)	When $z_i < 0 \to$ clockwise rotation ($\theta < 0$)
When $y_i \geq 0 \to$ clockwise rotation ($\theta < 0$)	When $z_i \geq 0 \to$ counterclockwise rotation ($\theta > 0$)

5.1 Elementary Functions Available Through CORDIC

The possible functions that could be obtained directly using CORDIC are listed in Table 5 for vectoring mode and in Table 6 for rotation mode. After n iterations, the output values of x, y, and z are determined by the operation performed by CORDIC. Table 5 summarizes the values of the outputs for the vectoring operation ($y \to 0$). Notice that we can obtain the square-root, $\tan^{-1}$, $\tanh^{-1}$ as well as number division. So now we have a way to divide numbers using hardware.

Table 6 summarizes the values of the outputs for the rotation operation ($z \to 0$). Notice that we can obtain

Table 5: CORDIC processor *true* output in the vectoring operation ($y \to 0$). x_0, y_0 and z_0 are the initial or input values to the CORDIC algorithm.

	x_n	z_n
$m = 1$	$\sqrt{x_0^2 + y_0^2}$	$z_0 + \tan^{-1}(y_0/x_0)$
$m = 0$	x_0	$z_0 + y_0/x_0$
$m = -1$	$\sqrt{x_0^2 - y_0^2}$	$z_0 + \tanh^{-1}(y_0/x_0)$

Table 6: CORDIC processor *true* output in the rotation operation ($z \to 0$). x_0, y_0 and z_0 are the initial or input values to the CORDIC algorithm.

	x_n	y_n
$m = 1$	$x_0 \cos z_0 - y_0 \sin z_0$	$y_0 \cos z_0 + x_0 \sin z_0$
$m = 0$	x_0	$y_0 + x_0 z_0$
$m = -1$	$x_0 \cosh z_0 + y_0 \sinh z_0$	$y_0 \cosh z_0 + x_0 \sinh z_0$

$\sin$, $\cos$, $\sinh$, $\cosh$ as well as number multiplication. Indirectly, we can obtain the exponential function e by proper choice of the mode ($m = -1$) and the initial values of x_0 and y_0.

Other elementary functions could also be obtained indirectly such as *exp*, $\sin^{-1}$, $\tan$, $\ln$ etc.

Table 7 shows the directly computable functions.

Table 7: CORDIC directly computable functions

$\sin z$	$\cos z$	$\tan^{-1} z$
$\sinh z$	$\cosh z$	$\tanh^{-1} z$
x/y	xz	$\tan^{-1}(y/z)$
$\sqrt{x^2 + y^2}$	$\sqrt{x^2 - y^2}$	e^z

CORDIC can also obtain more functions by repeated application of the algorithm. Table 8 shows the indirectly computable functions.

Table 8: CORDIC indirectly computable functions

$\tan z = \sin z / \cos z$	$\tanh z = \sinh z / \cosh z$
$\sin^{-1} w = \tan^{-1} \frac{w}{\sqrt{1-w^2}}$	$\cos^{-1} w = \tan^{-1} \frac{\sqrt{1-w^2}}{w}$
x/y	xz
$\sqrt{x^2 + y^2}$	$\sqrt{x^2 - y^2}$

5.2 The Scale Factor

When CORDIC terminates, the final values x_{n-1} and y_{n-1} do not represent the true results. The true values of x and y are given by:

$$x_{true} = x_{n-1}K_m \tag{60}$$
$$y_{true} = y_{n-1}K_m \tag{61}$$

where the multiplicative scale factor K_m is given by:

$$\begin{aligned} K_m &= 1/\prod_{i=0}^{n-1}\sqrt{1+m2^{-2i}} \\ &= \begin{cases} \prod_{i=0}^{n-1}\cos\theta_i & m=1 \\ 1 & m=0 \\ \prod_{i=1}^{n-1}\cosh\theta_i & m=-1 \end{cases} \end{aligned} \tag{62}$$

Table 10 shows the values of the scale factor for different number of iterations n for circular and hyperbolic modes. Typically the descaled values of x and y are obtained using **another** CORDIC operation.

Table 9: Scale factor for different number of iterations n for circular and hyperbolic modes.

n	8	12	16	24	32
K_1	0.607259112299	0.607252959139	0.607252935103	0.607252935009	0.607252935009
K_{-1}	1.205124099153	1.205136310559	1.205136310559	1.205136358446	1.205136358446

5.3 Maximum rotation Range

Equation (53) represents the accumulation of the rotation angles based on rotating the vector at each iteration. The value of the angle at each rotation is $\mu_i\theta_i$ and the max-min range of the accumulated angles is given by:

$$\theta_{max} = \sum_{i=0}^{n-1}|\theta_i| \tag{63}$$
$$\theta_{min} = -\sum_{i=0}^{n-1}|\theta_i| \tag{64}$$

Table 11 shows the convergence range for the different modes.

Table 10 shows the values of the scale factor for different number of iterations n for circular and hyperbolic modes. Typically the descaled values of x and y are obtained using **another** CORDIC operation.

Table 10: Rotation angle range for different number of iterations n for circular and hyperbolic modes.

n	8	12	16	24	32
θ circular (rad)	1.7355	1.7428	1.7433	1.7433	1.7433
θ circular (deg)	99.4353	99.8550	99.8812	99.8830	99.8830
θ hyper	1.0477	1.0550	1.0555	1.0555	1.0555

Table 11: Range of rotation angle θ for the three CORDIC modes.

m	θ_{max}
$m = 1$	1.74 radians = 99.69°
$m = 0$	$1 - 2^{-n}$
$m = -1$	1.13

REFERENCES

1. F. Gebali, T. Sui and A. Rayhan, "HCORDIC: A High-Radix Adaptive CORDIC Algorithm", Canadian Journal of Electrical & Computer Engineering, volume 25, number 4, 2000.

A Proving CORDIC Circular Mode

Let us write the update equation of CORDIC in the form

$$\begin{aligned} x' &= x_0 - y_0 \tan\theta && (65)\\ y' &= y_0 + x_0 \tan\theta && (66)\\ z' &= z_0 - \theta && (67) \end{aligned}$$

where we express $\delta = \tan\theta$.

A.1 Vectoring Mode ($y \to 0$)

In the vectoring mode, our objective is to have $y' = 0$. We can do that in one step if we choose:

$$\tan\theta = -\frac{y_0}{x_0} \tag{68}$$

Substituting in the update equations, we get:

$$\begin{aligned} x' &= \frac{x_0^2 + y_0^2}{x_0} = \frac{r^2}{x_0} && (69)\\ 0 &= y_0 + x_0 \tan\theta && (70)\\ z' &= z_0 + \tan^{-1}(y_0/x_0) && (71) \end{aligned}$$

The true value of x' is obtained after multiplying by the scale factor $\cos\theta$ and we get:

$$x'_{true} = \cos\theta \times \frac{r^2}{x_0} = r \tag{72}$$

Equations (71) and (72) produce the values of CORDIC in the first row ($m = 1$) in Table 5.

A.2 Rotation Mode ($z \to 0$)

In the rotation mode, our objective is to have $z' = 0$. We can do that in one step if we choose:

$$\theta = z_0 \tag{73}$$

Substituting in the update equations, we get:

$$\begin{aligned} x' &= x_0 - y_0 \tan\theta && (74)\\ y' &= y_0 + x_0 \tan\theta && (75)\\ 0 &= z_0 - \theta && (76) \end{aligned}$$

The true values of x' and y' are obtained after multiplying by the scale factor $\cos\theta$ and we get:

$$\begin{aligned} x'_{true} &= x_0 \cos z_0 - y_0 \sin z_0 && (77)\\ y'_{true} &= y_0 \cos z_0 + x_0 \sin z_0 && (78) \end{aligned}$$

Equations (77) and (78) produce the values of CORDIC in the first row ($m = 1$) in Table 6.

B Proving CORDIC Linear Mode

Let us write the update equation of CORDIC in the form

$$x' = x_0 \tag{79}$$
$$y' = y_0 + x_0\,\delta \tag{80}$$
$$z' = z_0 - \delta \tag{81}$$

B.1 Vectoring Mode ($y \to 0$)

In the vectoring mode, our objective is to have $y' = 0$. We can do that in one step if we choose:

$$\delta = -\frac{y_0}{x_0} \tag{82}$$

Substituting in the update equations, we get:

$$x' = x_0 \tag{83}$$
$$0 = y_0 + x_0\,\delta \tag{84}$$
$$z' = z_0 + y_0/x_0 \tag{85}$$

Equation (85) produces the values of CORDIC in the second row ($m = 0$) in Table 5.

B.2 Rotation Mode ($z \to 0$)

In the rotation mode, our objective is to have $z' = 0$. We can do that in one step if we choose:

$$\delta = z_0 \tag{86}$$

Substituting in the update equations, we get:

$$x' = x_0 \tag{87}$$
$$y' = y_0 + x_0 z_0 \tag{88}$$
$$0 = z_0 - \delta \tag{89}$$

Equation (88) produces the values of CORDIC in the second row ($m = 0$) in Table 6.

C Proving CORDIC Hyperbolic Mode

Let us write the update equation of CORDIC in the form

$$x' = x_0 + y_0\,\delta \tag{90}$$
$$y' = y_0 + x_0\,\delta \tag{91}$$
$$z' = z_0 - \theta \tag{92}$$

where we express $\delta = \tanh\theta$.

C.1 Vectoring Mode ($y \to 0$)

In the vectoring mode, our objective is to have $y' = 0$. We can do that in one step if we choose:

$$\delta = -\frac{y_0}{x_0} \tag{93}$$

Substituting in the update equations, we get:

$$\begin{aligned} x' &= \frac{x_0^2 - y_0^2}{x_0} = \frac{r^2}{x_0} && (94)\\ 0 &= y_0 + x_0 \times \frac{y_0}{x_0} && (95)\\ z' &= z_0 + \tanh^{-1}(y_0/x_0) && (96) \end{aligned}$$

The true value of x' is obtained after multiplying by the scale factor $\cos\theta$ and we get:

$$x'_{true} = \cosh\theta \times \frac{r^2}{x_0} = r \tag{97}$$

Equations (96) and (97) produce the values of CORDIC in the third row ($m = -1$) in Table 5.

C.2 Rotation Mode ($z \to 0$)

In the rotation mode, our objective is to have $z' = 0$. We can do that in one step if we choose:

$$\theta = z_0 \tag{98}$$

Substituting in the update equations, we get:

$$\begin{aligned} x' &= x_0 + y_0 \tanh\theta && (99)\\ y' &= y_0 + x_0 \tanh\theta && (100)\\ 0 &= z_0 - \theta && (101) \end{aligned}$$

The true values of x' and y' are obtained after multiplying by the scale factor $\cosh\theta$ and we get:

$$\begin{aligned} x'_{true} &= x_0\cosh z_0 + y_0 \sinh z_0 && (102)\\ y'_{true} &= y_0\cosh z_0 + x_0 \sinh z_0 && (103) \end{aligned}$$

Equations (102) and (103) produce the values of CORDIC in the third row ($m = -1$) in Table 6.