# A highly configurable 1st quadrant CORDIC core

in verilog

*Author: Dale Drinkard*
*dale@fpgadesigner.com*

**Rev .0.1**
**September 13, 2008**

## REVISION HISTORY

| Revision | Date | Author | Description |
|---|---|---|---|
| 0.1 | 09/13/08 | Dale Drinkard | First Draft |

## Table of Contents

## 1 CORDIC Overview

CORDIC (Coordinate Rotation Digital Computer) is an algorithm for computing transcendental functions like sine, cosine and arctangent. The method can also be easily extended to compute square roots as well as hyperbolic functions.

The algorithm works by reducing the calculation into a number of micro-rotations for which the arctangent value is precomputed and loaded in a table. This method reduces the computation to addition, subtraction, compares, and shifts. All functions easily performed by FPGAs.

## 2 CORDIC Theory

To rotate the coordinate ( $X_i, Y_i$ ) to the point ( $X_j, Y_j$ ) We use the following matrix calculation.

$$\begin{bmatrix} X_j \\ Y_j \end{bmatrix} = \begin{bmatrix} \cos\Theta & -\sin\Theta \\ \sin\Theta & \cos\Theta \end{bmatrix} * \begin{bmatrix} X_i \\ Y_i \end{bmatrix} \qquad (1)$$

Equation 1 can be modified by factoring a $\cos\Theta$ leading to the following:

$$\begin{bmatrix} X_j \\ Y_j \end{bmatrix} = \cos\Theta * \begin{bmatrix} 1 & -\tan\Theta \\ \tan\Theta & 1 \end{bmatrix} * \begin{bmatrix} X_i \\ Y_i \end{bmatrix} \qquad (2)$$

Any angle $\Theta$ in the first quadrant can be represented by the following equation.

$$\Theta_n = \sum_{n=0}^{\infty} S_i * \arctan\left(1/2^n\right) \qquad (3)$$

where $S_i$ = 0 or 1

For example, the angle 0 is computed by letting S_i in equation (3) be simply $S_i = 0,0,0,0\ldots$ . The angle $\Theta = \pi/2$ is computed by letting $S_i = 1,1,1,1\ldots$ . $\pi/4$ would be $1,0,0,0\ldots$ , and so on. In practice, due to the accuracy limits of the digital representation, the summation can be limited to the number of bits representing the value, or fewer. The $\arctan\left(1/2^n\right)$ quickly approaches 0.

Note that equation 3 can also be constructed such that S_i = -1 or 1. For example, the angle $\pi/4$ would be computed with the sequence $1,1,-1,-1,-1,\ldots$ . The advantage of this method will become apparent.

This results in the following equation for $\tan\Theta_n$

$$\tan\Theta_n = S_n * (1/2^n) \qquad (4)$$

Turning our attention back to equation (2), and given that the angle $\Theta$ can be represented as a weighted sum (3), we can represent the rotation from $(X_i, Y_i)$ to $(X_j, Y_j)$ as a sequence of smaller rotations where the $n+1$ rotational coordinate is derived from the $n^{th}$ coordinate using the following equation

$$\begin{bmatrix} X_{n+1} \\ Y_{n+1} \end{bmatrix} = \cos\Theta_n * \begin{bmatrix} 1 & -\tan\Theta_n \\ \tan\Theta_n & 1 \end{bmatrix} * \begin{bmatrix} X_n \\ Y_n \end{bmatrix} \quad (5)$$

By substituting equation (4) into equation (5) we replace the tangent function with a divide by 2^n , which is a simple shift operation

$$\begin{bmatrix} X_{n+1} \\ Y_{n+1} \end{bmatrix} = \cos\Theta_n * \begin{bmatrix} 1 & -S_n*(1/2^n) \\ S_n*(1/2^n) & 1 \end{bmatrix} * \begin{bmatrix} X_n \\ Y_n \end{bmatrix} \quad (6)$$

By implementing the algorithm letting S_n take on the values -1 and 1, and by noting that

$$\cos(\Theta) = \cos(-\Theta) \quad (7)$$

the $\cos(\Theta)$ term reduces to a constant K given by the equation

$$K = \prod_{n=0}^{\infty} \left(\cos(\arctan(1/2^n))\right) \approx 0.607253 \qquad (8)$$

The constant K means that the CORDIC function as an overall gain by the factor 1/K which is often referred to as the CORDIC gain. Since this is a constant, it can be taken into account in other parts of the system.

## 3 Core Description

The *highly configurable CORDIC core* realizes a 1st quadrant *coordinate rotational digital computer* algorithm to compute transcendental functions. The core, through `defines in the source, can realize RTL in one of three architectures:

Combinatorial
Iterative
Pipelined

The combinatorial realization solves equations in one clock cycle at the expense of many levels of logic. The iterative approach breaks the problem down into a number of iterations. This method reduces the amount of logic at the expense of needing several clock cycles to complete. Finally, the pipeline realization takes several clock cycles to complete, but the core can produce a new result on every clock cycle.

The CORDIC algorithm operates in one of two modes:

Rotate
Vector

In rotate mode the user feeds the core an angle an it computes the sin and cos.  In vector mode the inputs are sin/cos and the computer returns the arctangent.

In addition to core architecture, the core also supports computing angles in either radian or degree format.  There are also provisions for handling arbitrary bit lengths and iterations.  A full description of the various `defines can be found in the source.

The highly configurable 1st quadrant CORDIC core operates in the first quadrant only.  However, it is easy to extend this operation to the full circle by adding a coarse rotation to the front and back of the core.  For example, to compute the sin and cos of an angle in the 2nd quadrant, subtract $\pi/2$ from the angle, run through the core, and flip the sign of the y (sin) output.

## 4 Core Implementation

The core was verified in a Lattice ECP2-50 FPGA.  Table 1 shows the resultant performance and LUT utilization;

| Configuration | Combinatorial | Iterative | Pipeline |
|---|---|---|---|
| Slices | 597 | 235 | 562 |
| Fmax | 9mhz | 125mhz | 130mhz |

Table 1: utilization and performance in hardware

A functional testbench is also included.