

CORDIC Background (2A)

-
-

Copyright (c) 2010, 2011 Young W. Lim.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using OpenOffice and Octave.

CORDIC Background

1.A survey of CORDIC algorithms for FPGAs, Ray Andraka,
www.andraka.com/cordic.htm

Vector Rotation

$$x' = x \cos \phi + y \sin \phi$$

$$y' = y \cos \phi + x \sin \phi$$

$$x' = \cos \phi \cdot [x - y \tan \phi]$$

$$y' = \cos \phi \cdot [y + x \tan \phi]$$

$$x_{i+1} = K_i \cdot [x_i - y_i \cdot d_i \cdot 2^{-i}]$$

$$y_{i+1} = K_i \cdot [y_i + x_i \cdot d_i \cdot 2^{-i}]$$

$$K_i = \cos(\tan^{-1} 2^{-1}) = \frac{1}{\sqrt{1 + 2^{-2i}}}$$

$$d_i = \pm 1$$

Restrict rotation angle

$$\tan \phi = \pm 2^{-i}$$

Multiplication

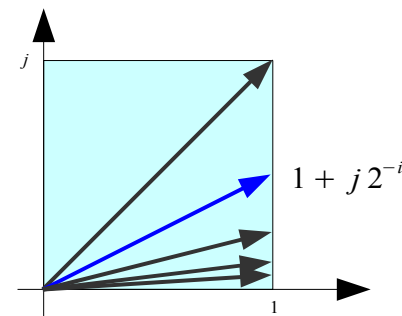
$$y \cdot \tan \phi$$

$$x \cdot \tan \phi$$

simple shift

$$\cos(\phi) = \cos(-\phi)$$

constant regardless
of direction



Vector Rotation

$$x_{i+1} = K_i \cdot [x_i - y_i \cdot d_i \cdot 2^{-i}]$$

$$y_{i+1} = K_i \cdot [y_i + x_i \cdot d_i \cdot 2^{-i}]$$

$$K_i = \cos(\tan^{-1} 2^{-i}) = \frac{1}{\sqrt{1 + 2^{-2i}}}$$

$$d_i = \pm 1$$

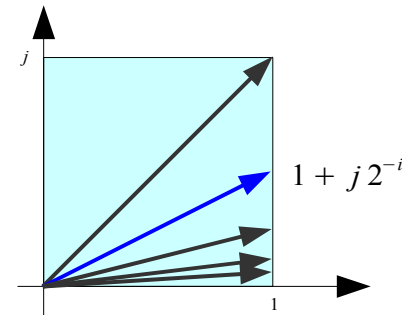
$$x_{i+1} = [x_i - y_i \cdot d_i \cdot 2^{-i}]$$

$$y_{i+1} = [y_i + x_i \cdot d_i \cdot 2^{-i}]$$

$$d_i = \pm 1$$

$$A_n = \prod_{i=1}^n K_i = \prod_{i=1}^n 1 / \sqrt{1 + 2^{-2i}}$$

$$A_n \rightarrow 1.647$$



Angle Accumulator

Rotation Mode

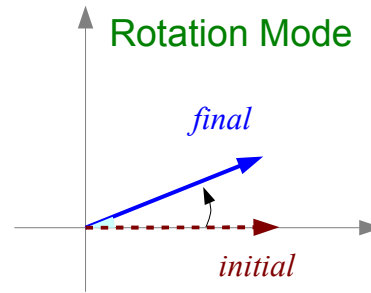
$$z_0 \leftarrow \phi \quad (\text{desired angle})$$

$$z_n \rightarrow 0$$

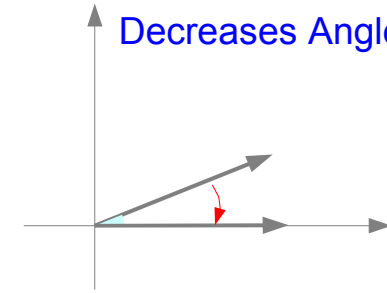
$$z_{i+1} = z_i - d_i \cdot \tan^{-1}(2^{-i})$$

$$d_i = -1 \quad \text{if } z_i < 0$$

$$d_i = +1 \quad \text{otherwise}$$



Angle Accumulator
Decreases Angle



Vectoring Mode

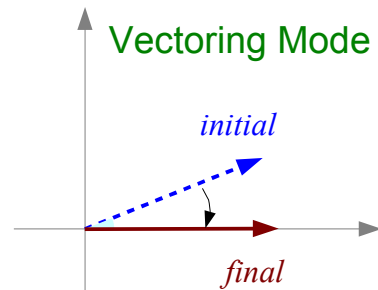
$$z_0 \leftarrow 0$$

$$z_n \rightarrow z_0 + \tan^{-1}(y_0/x_0)$$

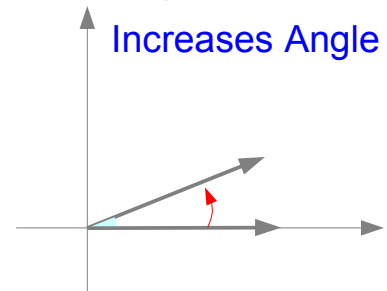
$$z_{i+1} = z_i - d_i \cdot \tan^{-1}(2^{-i})$$

$$d_i = +1 \quad \text{if } y_i < 0$$

$$d_i = -1 \quad \text{otherwise}$$

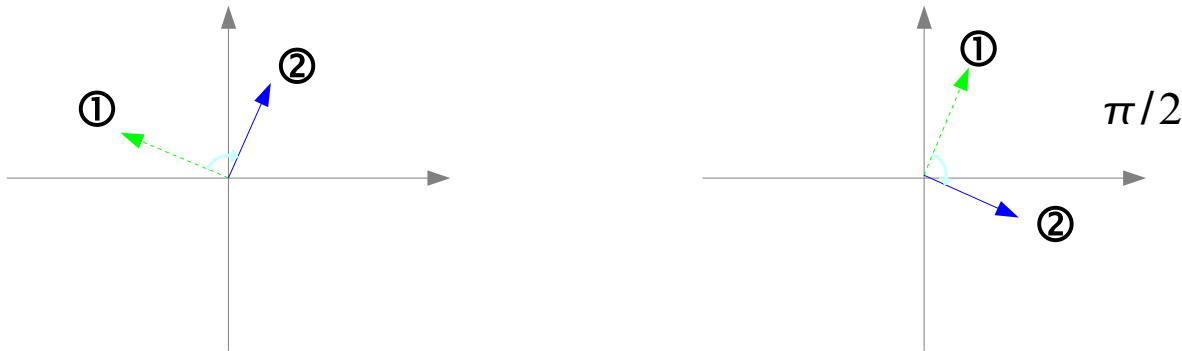


Angle Accumulator
Increases Angle



Initial Rotation $\pm\pi/2$

Positive Phase ($y > 0$) \rightarrow Rotate by -90 degrees



Negative Phase ($y < 0$) \rightarrow Rotate by $+90$ degrees



Resulting Phase \rightarrow $[-90, +90]$

$$\begin{aligned}x' &= -d \cdot y \\y' &= +d \cdot x \\z' &= z + d \cdot \frac{\pi}{2}\end{aligned}$$

$$\begin{aligned}d &= +1 \quad \text{if } y < 0 \\d &= -1 \quad \text{otherwise}\end{aligned}$$

No magnitude change

$$x' \leftarrow y$$

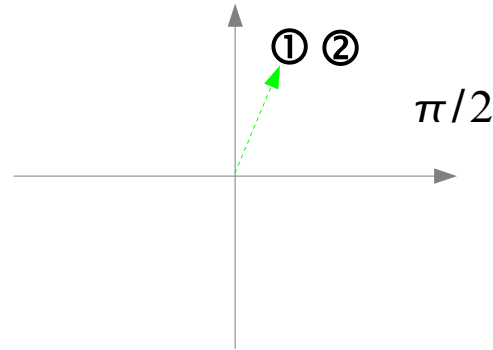
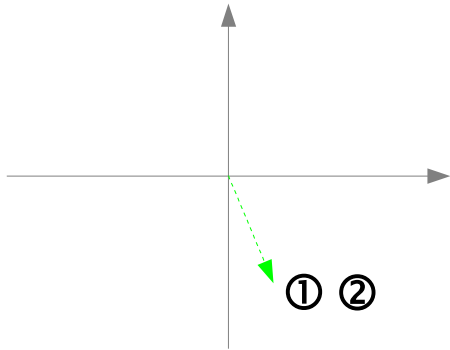
$$y' \leftarrow x$$

Consistent

Initial Rotation $0, +\pi$

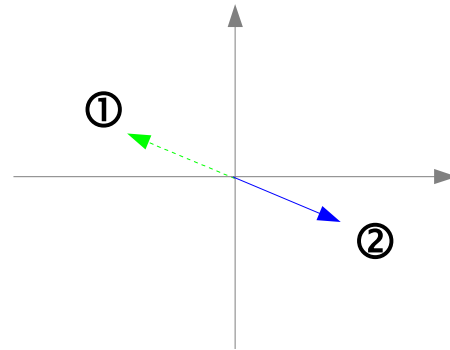
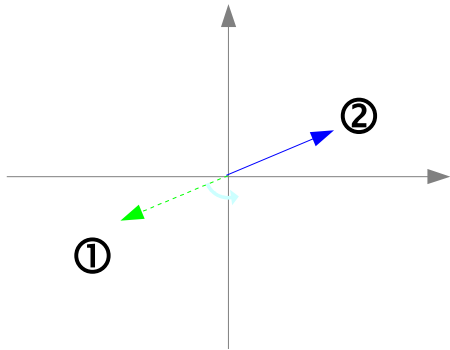
Positive x ($x > 0$)

→ Rotate by -90 degrees



Negative x ($x < 0$)

→ Rotate by $+90$ degrees



Resulting Phase



$[-90, +90]$

$$x' = +d \cdot x$$

$$y' = +d \cdot y$$

$$z' = z \quad \text{if } d = 1$$

$$z' = \pi - z \quad \text{if } d = -1$$

$$d = -1 \quad \text{if } x < 0$$

$$d = +1 \quad \text{otherwise}$$

No magnitude change

$$x' \leftarrow y$$

$$y' \leftarrow x$$

Convenient wiring in
FPGA

References

- [1] <http://en.wikipedia.org/>
- [2] CORDIC FAQ, www.dspguru.com