

# CORDIC in VHDL (1A)

---

- 
-

Copyright (c) 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](mailto:youngwlim@hotmail.com).

This document was produced by using OpenOffice and Octave.

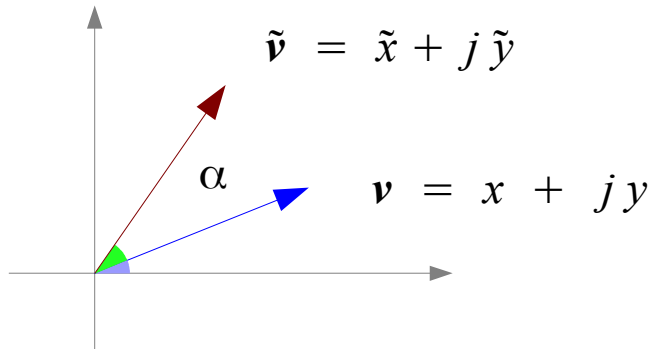
# CORDIC Background

---

1. G Hampson,  
A VHDL Implementation of a CORDIC Arithmetic Processor Chip  
Monash University, Technical Report 94-9, 1994

# Angle Expansion

$$\tilde{\mathbf{v}} = \mathbf{v} e^{j\alpha}$$



elementary angle

$$\alpha_0 = \tan^{-1}(2^0) =$$

$$\alpha_1 = \tan^{-1}(2^{-1}) =$$

$$\alpha_2 = \tan^{-1}(2^{-2}) =$$

$$\alpha_3 = \tan^{-1}(2^{-3}) =$$

$\alpha$  can be expanded by  
a set of elementary angles  $\alpha_i$   
pseudo-digits  $q_i$

$$\alpha_i \begin{cases} \pi/2 & i = -1 \\ \tan^{-1}(2^{-i}) & i = 0, 1, 2, \dots, n-1 \end{cases}$$

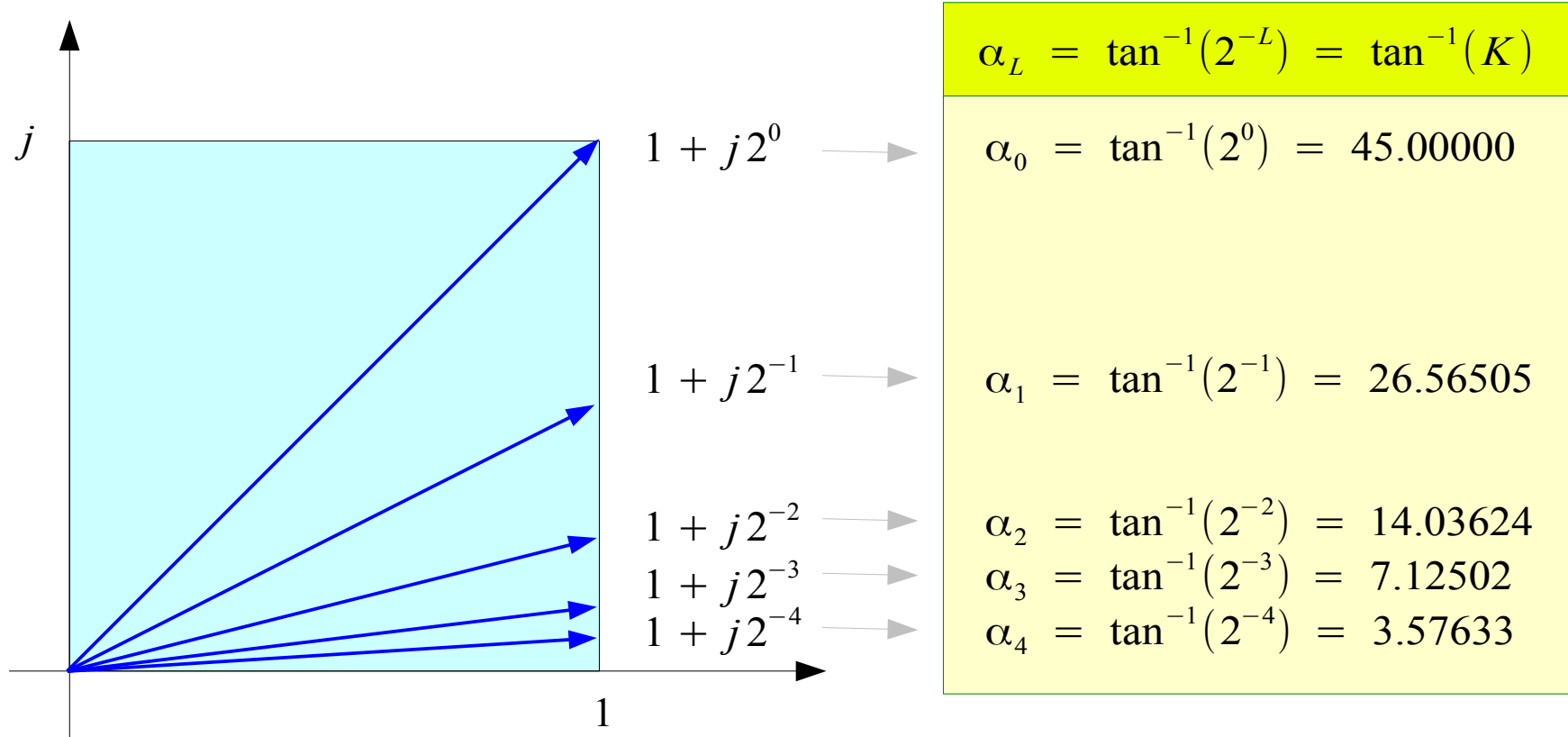
$$q_i \begin{cases} -1 \\ +1 \end{cases}$$

$$\alpha = \sum_{i=-1}^{n-1} q_i \cdot \alpha_i + z_n$$

angle expansion error  $z_n$

$$|z_n| \leq 2^{-(n-1)}$$

# Elementary Angle: $\tan^{-1}(K)$



Represent arbitrary angle  $\alpha$

in terms of  $\pm\alpha_0, \pm\alpha_1, \pm\alpha_2, \pm\alpha_3, \dots, \pm\alpha_L, \dots$   $\left(K = \frac{1}{2^L}, L = 0, 1, 2, \dots\right)$

# Phase and Magnitude of $1 + jK$ (1)

Cumulative Magnitude

$L$	$K = \frac{1}{2^L}$	$R = 1 + jK$	Phase of $R$	Magnitude of $R$	CORDIC Gain
0	1.0	$1 + j1.0$	$45^\circ$	1.41421356	1.414213562
1	0.5	$1 + j0.5$	$26.56505^\circ$	1.11803399	1.581138830
2	0.25	$1 + j0.25$	$14.03624^\circ$	1.03077641	1.629800601
3	0.125	$1 + j0.125$	$7.12502^\circ$	1.00778222	1.642484066
4	0.0625	$1 + j0.0625$	$3.57633^\circ$	1.00195122	1.645688916
5	0.03125	$1 + j0.03125$	$1.78991^\circ$	1.00048816	1.646492279
6	0.015625	$1 + j0.015625$	$0.89517^\circ$	1.00012206	1.646693254
7	0.007813	$1 + j0.007813$	$0.44761^\circ$	1.00003052	1.646743507
...	...	...	...	...	...
					1.647 ←

$$R = 1 + jK \xrightarrow[L = 0, 1, 2, \dots]{K = 1/2^L} \sqrt{1^2 + K^2} > 1.0$$

↓

# Rotating Vector

$$\tilde{\mathbf{v}} = \mathbf{v} e^{j\alpha}$$

$$= \mathbf{v} \exp\left(j\left(\sum_{i=-1}^{n-1} q_i \cdot \alpha_i + z_n\right)\right)$$

$$= \mathbf{v} \cdot \left(\prod_{i=-1}^{n-1} e^{jq_i \alpha_i}\right) \cdot e^{jz_n}$$

$$= \mathbf{v} \cdot (jq_{-1}) \left(\prod_{i=0}^{n-1} e^{jq_i \alpha_i}\right) \cdot e^{jz_n}$$

$$= \mathbf{v} \cdot (jq_{-1}) \left(\prod_{i=0}^{n-1} \cos(\alpha_i) \cdot (1 + jq_i 2^{-i})\right) \cdot e^{jz_n}$$

$$= \mathbf{v} \cdot (jq_{-1}) \left(\prod_{i=0}^{n-1} \cos(\alpha_i)\right) \cdot \left(\prod_{i=0}^{n-1} (1 + jq_i 2^{-i})\right) \cdot e^{jz_n}$$

$$\alpha = \sum_{i=-1}^{n-1} q_i \cdot \alpha_i + z_n$$

$$e^{j\theta} = \cos\theta + j\sin\theta$$

$$e^{jq_{-1}\alpha_0} = e^{jq_{-1}\frac{\pi}{2}} = jq_{-1} \quad (e^{\pm j\frac{\pi}{2}} = \pm j)$$

$$\begin{aligned} e^{jq_i \alpha_i} &= \cos(q_i \alpha_i) + j\sin(q_i \alpha_i) \\ &= \cos(q_i \alpha_i) \cdot (1 + j\tan(q_i \alpha_i)) \\ &= \cos(q_i \alpha_i) \cdot (1 + jq_i 2^{-i}) \\ &= \cos(\alpha_i) \cdot (1 + jq_i 2^{-i}) \end{aligned}$$

$$(\cos(\pm\alpha_i) = \cos(\alpha_i))$$



# Rotating Via Elementary Angles

$$\tilde{\mathbf{v}} = \mathbf{v} e^{j\alpha} \quad \text{rotated by } \alpha$$

$$\alpha = \sum_{i=-1}^{n-1} q_i \cdot \alpha_i + z_n$$

$$= \mathbf{v} \cdot \left( \prod_{i=0}^{n-1} \cos(\alpha_i) \right) \cdot (jq_{-1}) \cdot \left( \prod_{i=0}^{n-1} (1 + jq_i 2^{-i}) \right) \cdot e^{jz_n}$$



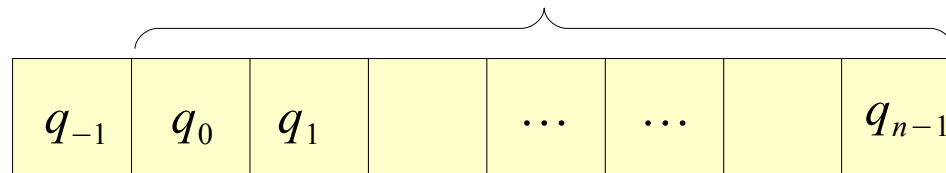
$$K_n = \prod_{i=0}^{n-1} \frac{1}{\sqrt{1 + 2^{-2i}}}$$

series rotations of  $\alpha_i$

Angle Expansion Error

$$\alpha - \sum_{i=-1}^{n-1} q_i \cdot \alpha_i = z_n$$

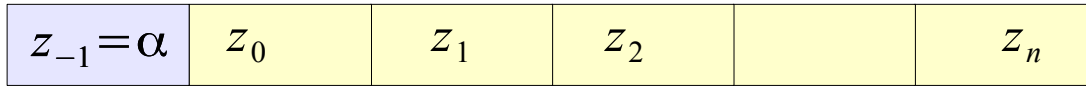
n iterations





# Angle Expansion

$$((((\alpha - q_{-1}\alpha_{-1}) - q_0\alpha_0) - q_1\alpha_1) \cdots - q_{n-1}\alpha_{n-1})$$



$$z_{i+1} = z_i - q_i \alpha_i$$

$$\begin{aligned} \text{if } (z_i \geq 0) & q_i = +1 \\ \text{if } (z_i < 0) & q_i = -1 \end{aligned}$$

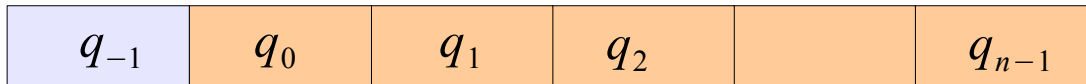
$$\begin{aligned} \text{if } (z_0 \geq 0) & q_0 = +1 & z_1 &= z_0 - \alpha_0 \\ \text{if } (z_0 < 0) & q_0 = -1 & z_1 &= z_0 + \alpha_0 \end{aligned}$$

$$\begin{aligned} \text{if } (z_1 \geq 0) & q_1 = +1 & z_2 &= z_1 - \alpha_1 \\ \text{if } (z_1 < 0) & q_1 = -1 & z_2 &= z_1 + \alpha_1 \end{aligned}$$

$$\begin{aligned} \text{if } (z_2 \geq 0) & q_2 = +1 & z_3 &= z_2 - \alpha_2 \\ \text{if } (z_2 < 0) & q_2 = -1 & z_3 &= z_2 + \alpha_2 \end{aligned}$$

$$\begin{aligned} \text{if } (z_{-1} \geq 0) & q_{-1} = +1 \\ \text{if } (z_{-1} < 0) & q_{-1} = -1 \end{aligned}$$

$$\begin{aligned} \text{if } (z_{n-1} \geq 0) & q_{n-1} = +1 & z_n &= z_{n-1} - \alpha_{n-1} \\ \text{if } (z_{n-1} < 0) & q_{n-1} = -1 & z_n &= z_{n-1} + \alpha_{n-1} \end{aligned}$$



# CORDIC Function (1)

$$\tilde{\mathbf{v}} = \mathbf{v} e^{j\alpha} \quad \text{rotated by } \alpha$$

Rotated vector after n iteration

$$\mathbf{v}_n = \mathbf{v} \cdot e^{j\alpha} \cdot e^{-jz_n}$$

$$\mathbf{v}_{i+1} = \mathbf{v}_i \cdot (1 + jq_i \cdot 2^{-i})$$

$$\begin{aligned} x_{i+1} + jy_{i+1} &= (x_i + jy_i) \cdot (1 + jq_i \cdot 2^{-i}) \\ &= (x_i - y_i \cdot q_i \cdot 2^{-i}) + j(y_i + x_i \cdot q_i \cdot 2^{-i}) \end{aligned}$$

$$\begin{cases} x_{i+1} = (x_i - y_i \cdot q_i \cdot 2^{-i}) \\ y_{i+1} = (y_i + x_i \cdot q_i \cdot 2^{-i}) \end{cases}$$

$$\alpha = \sum_{i=-1}^{n-1} q_i \cdot \alpha_i + z_n$$

$$\alpha - z_n = \sum_{i=-1}^{n-1} q_i \cdot \alpha_i$$

$$\mathbf{v}_0 = \mathbf{v}_{-1} \cdot (jq_{-1})$$

$$\begin{aligned} x_0 + jy_0 &= (x_{-1} + jy_{-1}) \cdot (jq_{-1}) \\ &= (-q_{-1} \cdot y_{-1}) + j(q_{-1} \cdot x_{-1}) \end{aligned}$$

$$\begin{cases} x_0 = -q_{-1} \cdot y_{-1} \\ y_0 = q_{-1} \cdot x_{-1} \end{cases}$$

# CORDIC Function (2)

$$\begin{aligned} \text{if } (z_i \geq 0) & \quad q_i = +1 \\ \text{if } (z_i < 0) & \quad q_i = -1 \end{aligned}$$

$$v_{i+1} = v_i \cdot (1 + j q_i \cdot 2^{-i})$$



$$\begin{cases} x_{i+1} = (x_i - y_i \cdot q_i \cdot 2^{-i}) \\ y_{i+1} = (y_i + x_i \cdot q_i \cdot 2^{-i}) \end{cases}$$

$$z_{i+1} = z_i - q_i \alpha_i$$

$$v_0 = v_{-1} \cdot (j q_{-1})$$



$$\begin{cases} x_0 = -q_{-1} \cdot y_{-1} \\ y_0 = q_{-1} \cdot x_{-1} \end{cases}$$

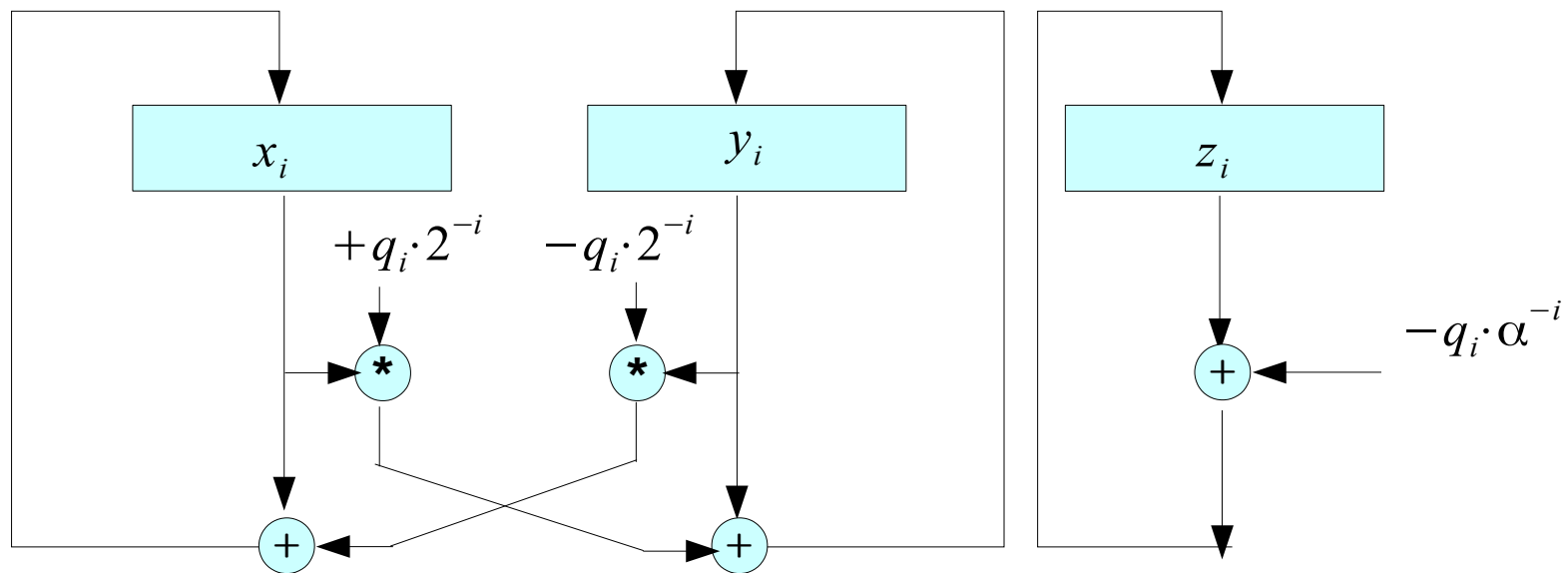
# CORDIC Datapath (1)

$$\begin{cases} \text{if } (z_i \geq 0) & q_i = +1 \\ \text{if } (z_i < 0) & q_i = -1 \end{cases}$$

$$\begin{cases} x_{i+1} = (x_i - y_i \cdot q_i \cdot 2^{-i}) \\ y_{i+1} = (y_i + x_i \cdot q_i \cdot 2^{-i}) \end{cases}$$

$$\begin{cases} x_0 = -q_{-1} \cdot y_{-1} \\ y_0 = q_{-1} \cdot x_{-1} \end{cases}$$

$$z_{i+1} = z_i - q_i \alpha_i$$



# Rotating Vector

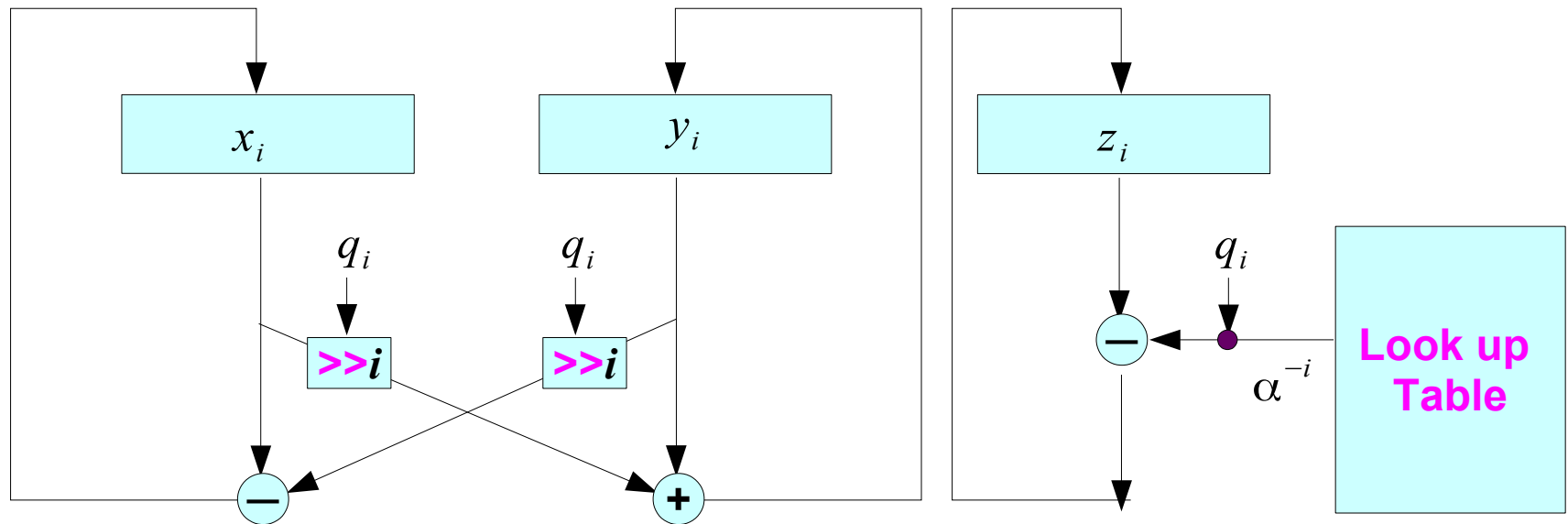
$$\begin{cases} \text{if } (z_i \geq 0) & q_i = +1 \\ \text{if } (z_i < 0) & q_i = -1 \end{cases}$$

! (msb of  $z_i$ )  $\rightarrow$   $q_i$

$$\begin{cases} x_{i+1} = (x_i - y_i \cdot q_i \cdot 2^{-i}) \\ y_{i+1} = (y_i + x_i \cdot q_i \cdot 2^{-i}) \end{cases}$$

$$\begin{cases} x_0 = -q_{-1} \cdot y_{-1} \\ y_0 = q_{-1} \cdot x_{-1} \end{cases}$$

$$z_{i+1} = z_i - q_i \alpha_i$$



# Rotating Vector

$$\tilde{\mathbf{v}} = \mathbf{v} e^{j\alpha} \quad \text{rotated by } \alpha$$

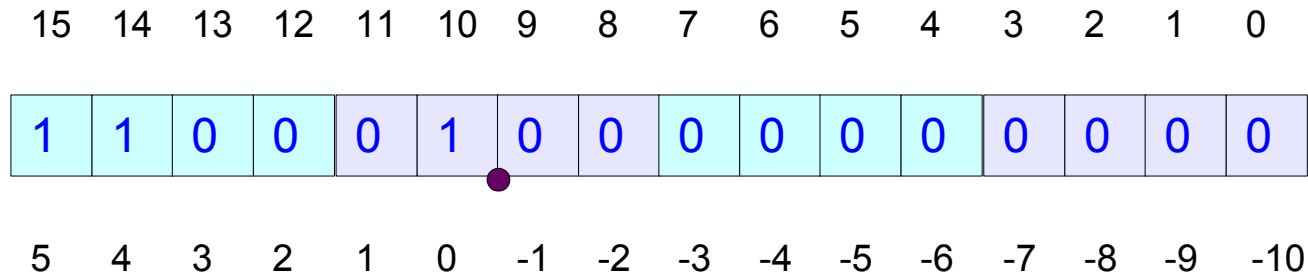
$$\alpha = \sum_{i=-1}^{n-1} q_i \cdot \alpha_i + z_n$$

$$= \mathbf{v} \cdot \left( \prod_{i=0}^{n-1} \cos(\alpha_i) \right) \cdot (jq_{-1}) \cdot \left( \prod_{i=0}^{n-1} (1 + jq_i 2^{-i}) \right) \cdot e^{jz_n}$$

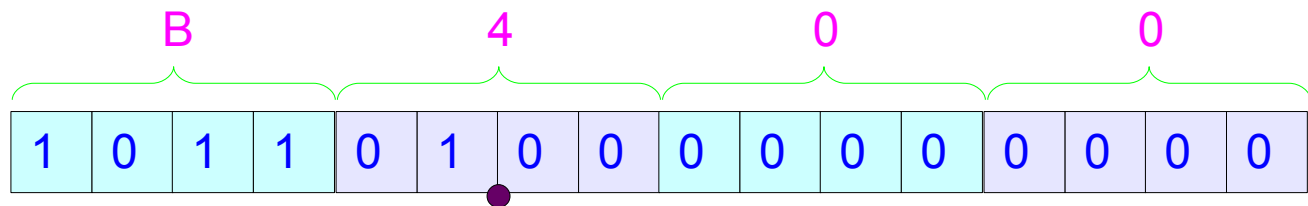
$q_{-1}$	$q_0$	$q_1$		...	...		$q_{n-1}$
----------	-------	-------	--	-----	-----	--	-----------

# Binary Representation of Elementary Angles (1)

16-bit



$$2^5 + 2^4 + 2^0 = 32 + 16 + 1 = 49$$



$$2^5 + 2^3 + 2^2 + 2^0 = 32 + 8 + 4 + 1 = 45$$

$$B400_{16} / 1024_{10} = 45_{10}$$

# Binary Representation of Elementary Angles (2)

i	angle	Hex	Dec / 1024	Degree	16-bit Binary Number
0	atan(2 <sup>0</sup> )	B400	46080 / 1024	45.0000	1011_0100_0000_0000
1	atan(2 <sup>-1</sup> )	6A43	27203 / 1024	26.5654	0110_1010_0100_0011
2	atan(2 <sup>-2</sup> )	3825	14373 / 1024	14.0361	0011_1000_0010_0101
3	atan(2 <sup>-3</sup> )	1C80	7296 / 1024	7.1250	0001_1100_1000_0000
4	atan(2 <sup>-4</sup> )	0E40	3648 / 1024	3.5625	0000_1110_0100_0000
5	atan(2 <sup>-5</sup> )	0729	1833 / 1024	1.7900	0000_0111_0010_1001
6	atan(2 <sup>-6</sup> )	0395	917 / 1024	0.8955	0000_0011_1001_0101
7	atan(2 <sup>-7</sup> )	01CA	458 / 1024	0.4473	0000_0001_1100_1010
8	atan(2 <sup>-8</sup> )	00E5	229 / 1024	0.2236	0000_0000_1110_0101
9	atan(2 <sup>-9</sup> )	0073	115 / 1024	0.1123	0000_0000_0111_0011
10	atan(2 <sup>-10</sup> )	0039	57 / 1024	0.0557	0000_0000_0011_1001
11	atan(2 <sup>-11</sup> )	001D	29 / 1024	0.0283	0000_0000_0001_1101
12	atan(2 <sup>-12</sup> )	000E	14 / 1024	0.0137	0000_0000_0000_1110
13	atan(2 <sup>-13</sup> )	0007	7 / 1024	0.0068	0000_0000_0000_0111
14	atan(2 <sup>-14</sup> )	0004	4 / 1024	0.0039	0000_0000_0000_0100
15	atan(2 <sup>-15</sup> )	0002	2 / 1024	0.0019	0000_0000_0000_0010
16	atan(2 <sup>-16</sup> )	0001	1 / 1024	0.0010	0000_0000_0000_0001



# Rounding – Reducing z update error

$$\hat{z}_i = 2^i \cdot z_i \quad \text{scaling by } 2^i$$

$$\begin{aligned}\hat{z}_{i+1} &= 2^{i+1} \cdot z_{i+1} \\ &= 2 \cdot 2^i \cdot (z_i - q_i \alpha_i) & z_{i+1} &= z_i - q_i \alpha_i \\ &= 2 \cdot (2^i z_i - q_i 2^i \alpha_i) & \hat{\alpha}_i &= 2^i \alpha_i \\ &= 2 \cdot (\hat{z}_i - q_i \hat{\alpha}_i)\end{aligned}$$

$$\begin{aligned}\hat{z}_{i+1} &= 2 \cdot (\hat{z}_i - q_i \hat{\alpha}_i) \\ \hat{\alpha}_i &= 2^i \alpha_i = 2^i \tan(2^{-i})\end{aligned}$$

## References

- [1] <http://en.wikipedia.org/>
- [2] G Hampson, A VHDL Implementation of a CORDIC Arithmetic Processor Chip  
Monash University, Technical Report 94-9, 1994