CORDIC in VHDL (1A)

-
-

Young Won Lim
11/10/2011

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

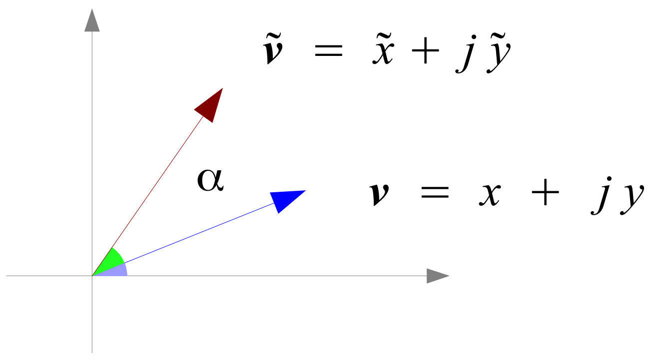
This document was produced by using OpenOffice and Octave.

CORDIC Background

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

Angle Expansion

$$\tilde{\boldsymbol{v}} = \boldsymbol{v}\, e^{j\alpha}$$



elementary angle

$$\alpha_0 = \tan^{-1}(2^{0}) =$$
$$\alpha_1 = \tan^{-1}(2^{-1}) =$$
$$\alpha_2 = \tan^{-1}(2^{-2}) =$$
$$\alpha_3 = \tan^{-1}(2^{-3}) =$$

α can be expanded by

a set of elementary angles α_i

pseudo-digits q_i

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

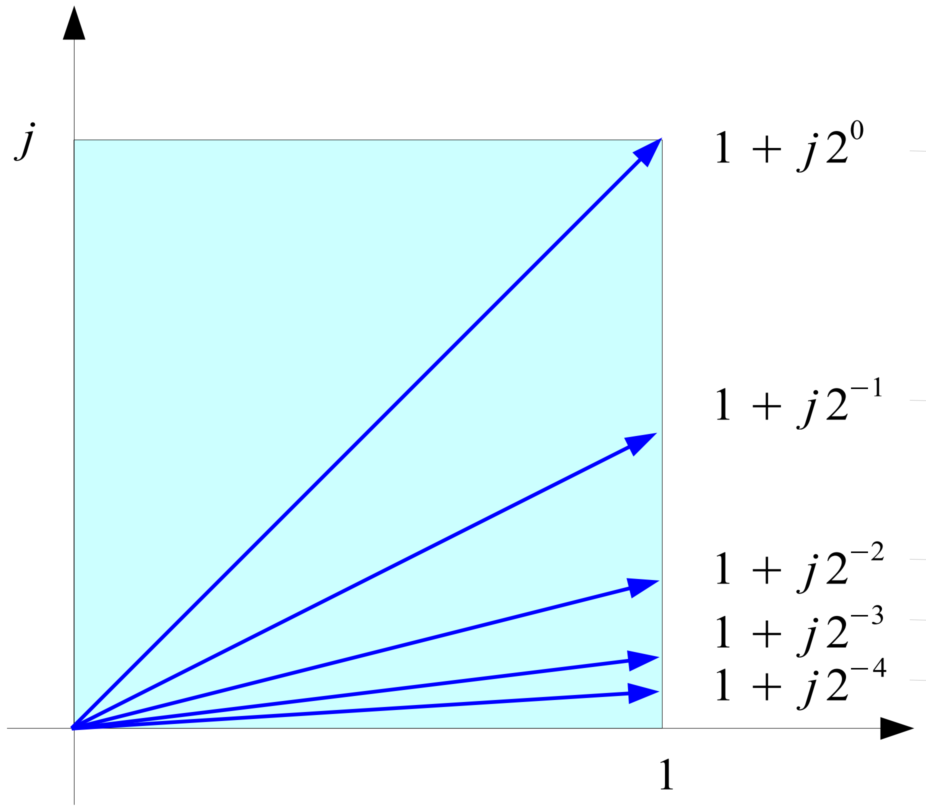
$$q_i \begin{cases} -1 \\ +1 \end{cases}$$

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

angle expansion error z_n

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

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



$1 + j2^{0}$ → $\alpha_0 = \tan^{-1}(2^{0}) = 45.00000$

$1 + j2^{-1}$ → $\alpha_1 = \tan^{-1}(2^{-1}) = 26.56505$

$1 + j2^{-2}$ → $\alpha_2 = \tan^{-1}(2^{-2}) = 14.03624$

$1 + j2^{-3}$ → $\alpha_3 = \tan^{-1}(2^{-3}) = 7.12502$

$1 + j2^{-4}$ → $\alpha_4 = \tan^{-1}(2^{-4}) = 3.57633$

$$\alpha_L = \tan^{-1}(2^{-L}) = \tan^{-1}(K)$$

Represent arbitrary angle α

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

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°	1.41421356	1.414213562
1	0.5	1 + j0.5	26.56505°	1.11803399	1.581138830
2	0.25	1 + j0.25	14.03624°	1.03077641	1.629800601
3	0.125	1 + j0.125	7.12502°	1.00778222	1.642484066
4	0.0625	1 + j0.0625	3.57633°	1.00195122	1.645688916
5	0.03125	1 + j0.03125	1.78991°	1.00048816	1.646492279
6	0.015625	1 + j0.015625	0.89517°	1.00012206	1.646693254
7	0.007813	1 + j0.007813	0.44761°	1.00003052	1.646743507
•••	•••	•••	•••	•••	•••
					1.647 ←

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

Rotating Vector

$$\tilde{\boldsymbol{v}} = \boldsymbol{v}\, e^{j\alpha}$$

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

$$= \boldsymbol{v} \cdot \left(\prod_{i=-1}^{n-1} e^{j q_i \alpha_i} \right) \cdot e^{j z_n}$$

$$= \boldsymbol{v} \cdot (j q_{-1}) \left(\prod_{i=0}^{n-1} e^{j q_i \alpha_i} \right) \cdot e^{j z_n}$$

$$= \boldsymbol{v} \cdot (j q_{-1}) \left(\prod_{i=0}^{n-1} \cos(\alpha_i) \cdot (1 + j\, q_i 2^{-i}) \right) \cdot e^{j z_n}$$

$$= \boldsymbol{v} \cdot (j q_{-1}) \left(\prod_{i=0}^{n-1} \cos(\alpha_i) \right) \cdot \left(\prod_{i=0}^{n-1} (1 + j\, q_i 2^{-i}) \right) \cdot e^{j z_n}$$

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

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

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

$$e^{j q_i \alpha_i} = \cos(q_i\, \alpha_i) + j \sin(q_i\, \alpha_i)$$

$$= \cos(q_i\, \alpha_i) \cdot (1 + j \tan(q_i\, \alpha_i))$$

$$= \cos(q_i\, \alpha_i) \cdot (1 + j\, q_i 2^{-i})$$

$$= \cos(\alpha_i) \cdot (1 + j\, q_i 2^{-i})$$

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

Rotating Via Elementary Angles

$$\tilde{\boldsymbol{v}} = \boldsymbol{v}\, e^{j\alpha}$$ rotated by α

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

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

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

series rotations of α_i

Angle Expansion Error

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

n iterations

q_{-1}	q_0	q_1		$\cdots$	$\cdots$		q_{n-1}

Angle Expansion

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

$z_{-1}=\alpha$	z_0	z_1	z_2		z_n

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

$$\begin{array}{ll} if\ (z_0 \ge 0) & q_0 = +1 \\ if\ (z_0 < 0) & q_0 = -1 \end{array} \qquad \begin{array}{l} z_1 = z_0 - \alpha_0 \\ z_1 = z_0 + \alpha_0 \end{array}$$

$$\begin{array}{ll} if\ (z_i \ge 0) & q_i = +1 \\ if\ (z_i < 0) & q_i = -1 \end{array}$$

$$\begin{array}{ll} if\ (z_1 \ge 0) & q_1 = +1 \\ if\ (z_1 < 0) & q_1 = -1 \end{array} \qquad \begin{array}{l} z_2 = z_1 - \alpha_1 \\ z_2 = z_1 + \alpha_1 \end{array}$$

$$\begin{array}{ll} if\ (z_2 \ge 0) & q_2 = +1 \\ if\ (z_2 < 0) & q_2 = -1 \end{array} \qquad \begin{array}{l} z_3 = z_2 - \alpha_2 \\ z_3 = z_2 + \alpha_2 \end{array}$$

$$\begin{array}{ll} if\ (z_{-1} \ge 0) & q_{-1} = +1 \\ if\ (z_{-1} < 0) & q_{-1} = -1 \end{array}$$

$$\begin{array}{ll} if\ (z_{n-1} \ge 0) & q_{n-1} = +1 \\ if\ (z_{n-1} < 0) & q_{n-1} = -1 \end{array} \qquad \begin{array}{l} z_n = z_{n-1} - \alpha_{n-1} \\ z_n = z_{n-1} + \alpha_{n-1} \end{array}$$

q_{-1}	q_0	q_1	q_2		q_{n-1}

CORDIC Function (1)

$$\tilde{\boldsymbol{v}} = \boldsymbol{v}\, e^{j\alpha}$$ rotated by α

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

Rotated vector after n iteration

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

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

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

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

$$x_{i+1} + jy_{i+1} = (x_i + jy_i) \cdot \left(1 + j\, q_i \cdot 2^{-i}\right)$$
$$= \left(x_i - y_i \cdot q_i \cdot 2^{-i}\right) + j\left(y_i + x_i \cdot q_i \cdot 2^{-i}\right)$$

$$x_0 + jy_0 = (x_{-1} + jy_{-1}) \cdot \left(j\, q_{-1}\right)$$
$$= \left(-q_{-1} \cdot y_{-1}\right) + j\left(q_{-1} \cdot x_{-1}\right)$$

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

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

CORDIC Function (2)

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

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

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

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

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

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

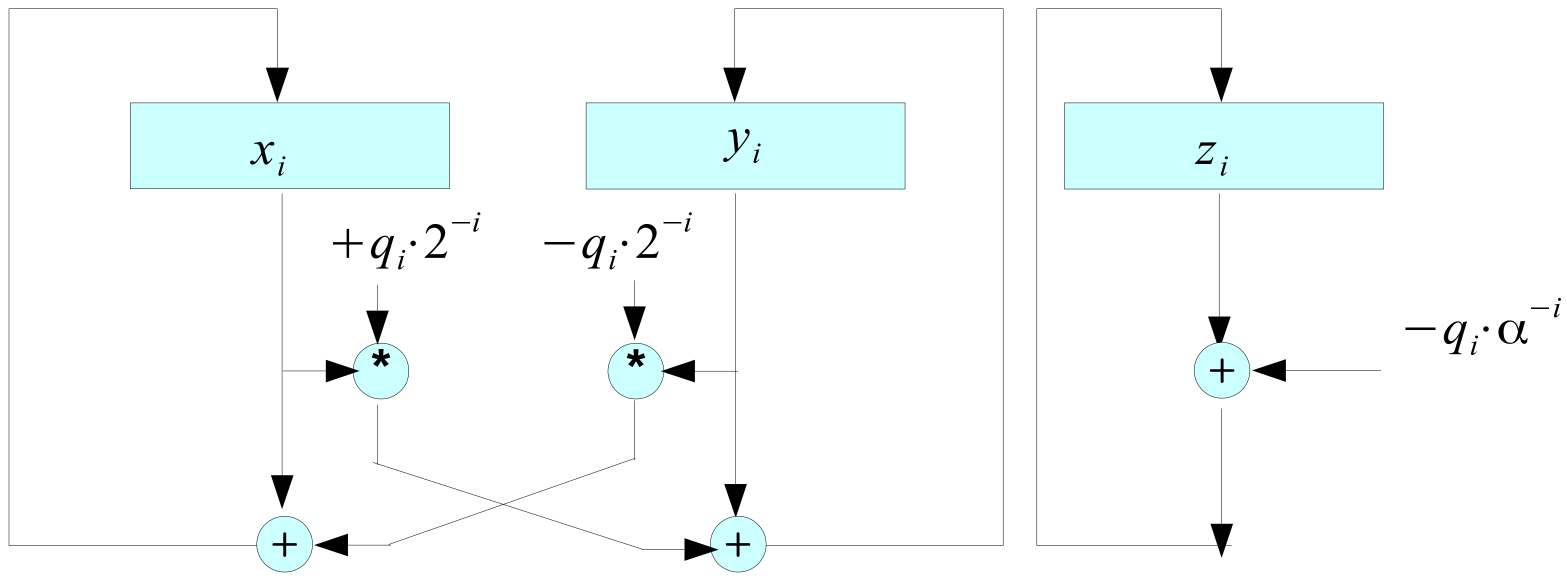
CORDIC Datapath (1)

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

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

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

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



Rotating Vector

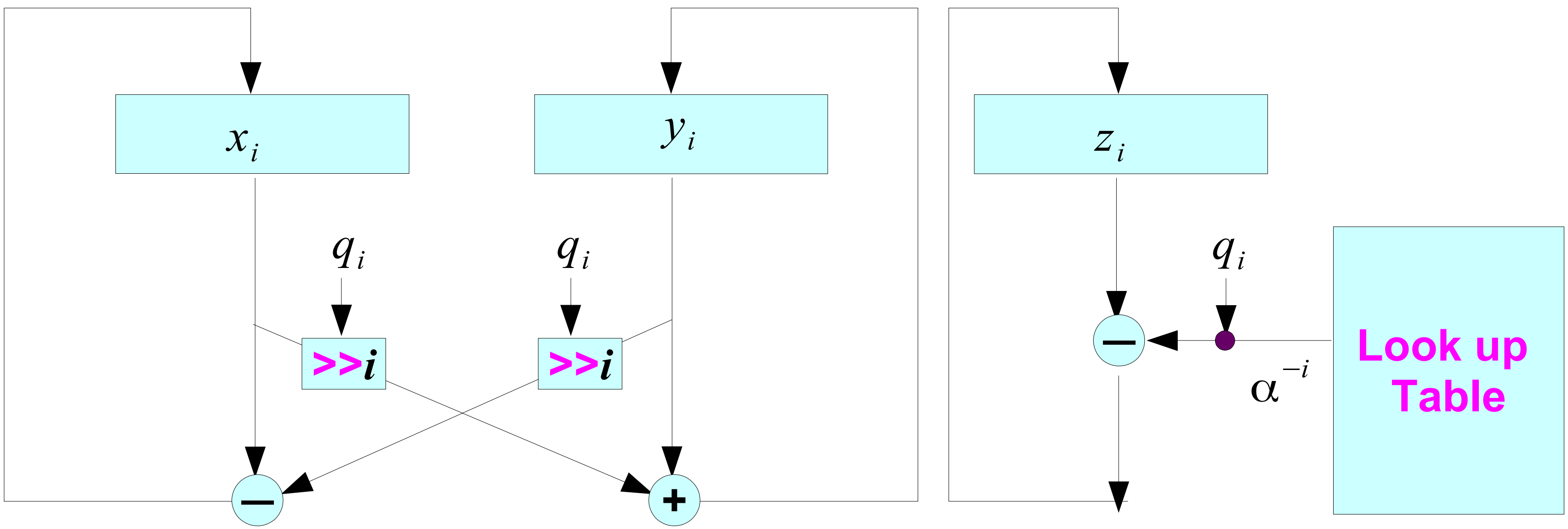
$$\begin{aligned} &if\ (z_i \geq 0) \quad q_i = +1 \\ &if\ (z_i < 0) \quad q_i = -1 \end{aligned}$$

! (msb of z_i) ➡ 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

$$\tilde{\boldsymbol{v}} = \boldsymbol{v}\, e^{j\alpha}$$ rotated by α

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

$$= \boldsymbol{v} \cdot \left(\prod_{i=0}^{n-1} \cos(\alpha_i) \right) \cdot (j q_{-1}) \cdot \left(\prod_{i=0}^{n-1} (1 + j\, q_i 2^{-i}) \right) \cdot e^{j z_n}$$

q_{-1}	q_0	q_1		$\cdots$	$\cdots$		q_{n-1}

Binary Representation of Elementary Angles (1)

16-bit

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0
5	4	3	2	1	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10

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

B				4				0				0			
1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0

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

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

Binary Representation of Elementary Angles (2)

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

CORDIC Accuracy (1)

$$\tilde{\boldsymbol{v}} = \boldsymbol{v}\, e^{j\alpha}$$ rotated by α

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

Rotated vector after n iteration

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

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

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

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

m: bus width of **v**

i-th stage

i ≤ m: $v_{i+1} \Leftarrow v_i \cdot 2^{-i}$

updated with the truncated value

$$x_{i+1} + jy_{i+1} = \left(x_i + jy_i\right) \cdot \left(1 + j\, q_i \cdot 2^{-i}\right)$$

$$= \left(x_i - y_i \cdot q_i \cdot 2^{-i}\right) + j\left(y_i + x_i \cdot q_i \cdot 2^{-i}\right)$$

i > m: $v_{i+1} \Leftarrow 0$

updated with the truncated value

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

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

CORDIC Accuracy (2)

$$\tilde{\boldsymbol{v}} = \boldsymbol{v}\, e^{j\alpha}$$ rotated by α

Rotated vector after n iteration

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

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

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

$$if\ (z_i \geq 0) \quad q_i = +1$$
$$if\ (z_i < 0) \quad q_i = -1$$

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

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

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

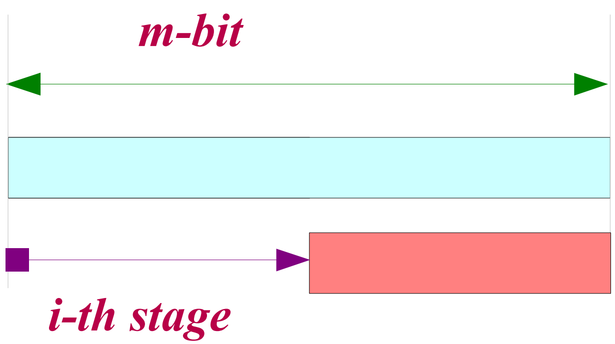
$$\begin{cases} x_{i+1} = (x_i - y_i \cdot q_i \cdot 2^{-i}) \\ y_{i+1} = (y_i + x_i \cdot q_i \cdot 2^{-i}) \end{cases}$$

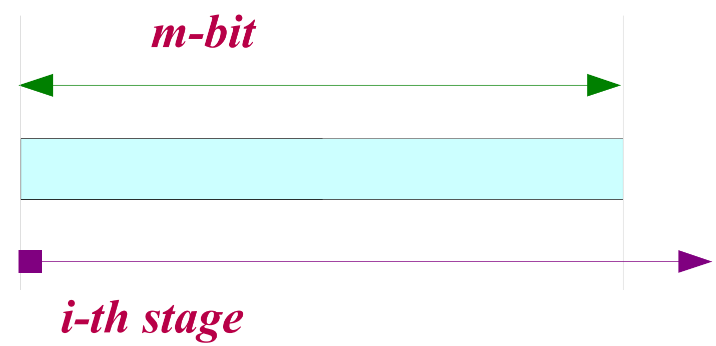
$$\begin{cases} x_0 = -q_{-1} \cdot y_{-1} \\ y_0 = q_{-1} \cdot x_{-1} \end{cases}$$

CORDIC Accuracy (3)

Truncation Error ⇐ Finite Word Length

Approximation Error ⇐ Finite Number of Iterations





$i \le m$:

v_{i+1} is updated
with the truncated value $v_i \cdot 2^{-i}$

$i > m$:

$v_{i+1} = v_i$ the update will be 0

CORDIC Accuracy (4)

How closely the input rotation angle was approximated
by the summation of elementary angles α_i

Error in v after n iterations $\propto$ Error in

increase in z datapath width ⇨ increase the accuracy of z update

⇨ increase the accuracy of v update

$\log n + 2$ extra bits ⇨ n bit accuracy after n iterations

4 bit accuracy after 4 iterations ⇦ $4 + \log 4 + 2 = 8$ bit datapath

8 bit accuracy after 8 iterations ⇦ $8 + \log 8 + 2 = 13$ bit datapath

16 bit accuracy after 16 iterations ⇦ $16 + \log 16 + 2 = 22$ bit datapath

Upper Bound

CORDIC Accuracy (5)

Rotational Mode

$\alpha \rightarrow 0$ By adding / subtracting sub rotational angles

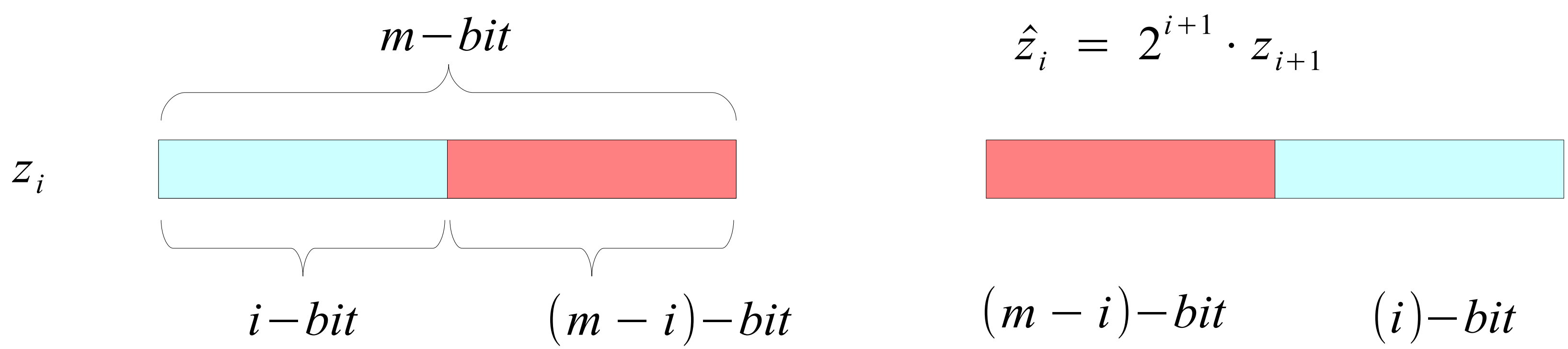
In the final iteration $z_i \rightarrow 0$

Z update error $z_i \approx 2^{-i}$ i-th stage

m-bit bus width

$(m-i)$ bits are used to represent error

$$\hat{z}_i = 2^{i+1} \cdot z_{i+1}$$



Rounding – Reducing z update error

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

$$\hat{z}_{i+1} = 2^{i+1} \cdot z_{i+1}$$

$$= 2 \cdot 2^i \cdot (z_i - q_i \alpha_i)$$

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

$$= 2 \cdot (2^i z_i - q_i 2^i \alpha_i)$$

$$\hat{\alpha}_i = 2^i \alpha_i$$

$$= 2 \cdot (\hat{z}_i - q_i \hat{\alpha}_i)$$

$$\hat{z}_{i+1} = 2 \cdot (\hat{z}_i - q_i \hat{\alpha}_i)$$

$$\hat{\alpha}_i = 2^i \alpha_i = 2^i \tan(2^{-i})$$

$$\hat{z}_i = 2^i \cdot z_i$$

$$\hat{\alpha}_i = 2^i \alpha_i$$

Truncation Error (1)

Positive Number + 45

45	2D	0010_1101
22	16	0001_0110
11	0B	0000_1011
05	05	0000_0101
02	02	0000_0010
01	01	0000_0001
00	00	0000_0000

Positive Number + 45 with rounding

45	2D	0010_1101	
23	17	0001_0111	1
12	0C	0000_1100	1
06	06	0000_0110	0
03	03	0000_0011	0
02	02	0000_0010	1
01	01	0000_0001	0

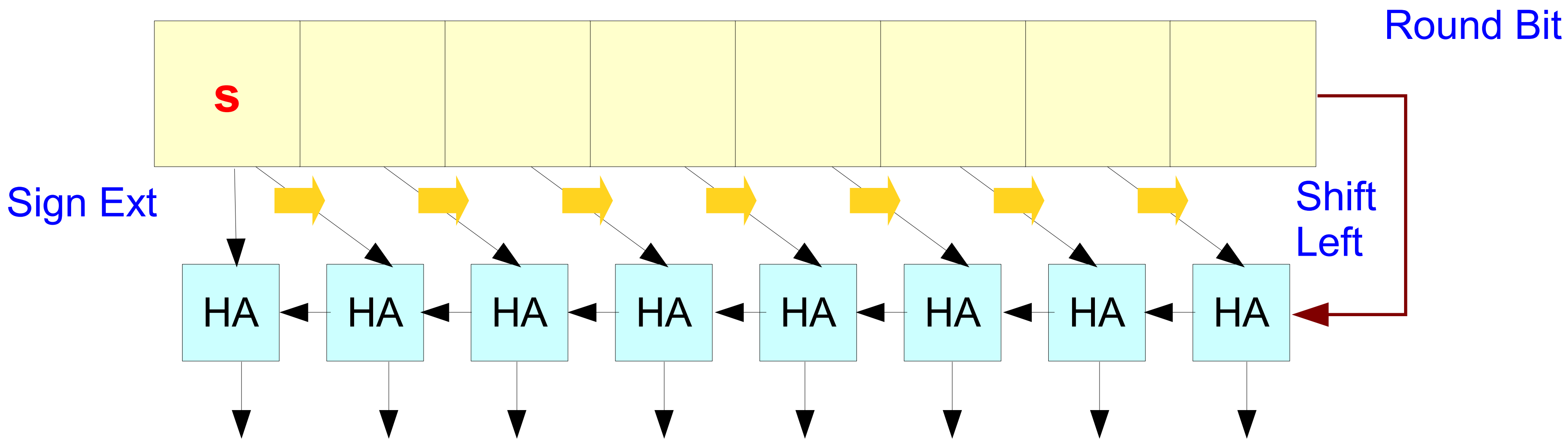
Negative Number – 45

-45	D3	1101_0011
-23	E9	1110_1001
-12	F4	1111_0100
-06	FA	1111_1010
-03	FD	1111_1101
-02	FE	1111_1110
-01	FF	1111_1111

Negative Number – 45 with rounding

45	2D	1101_0011	
-22	EA	1110_1010	1
-11	F5	1111_0101	0
-05	FB	1111_1011	1
-02	FD	1111_1110	1
-01	FF	1111_1111	0
00	00	0000_0000	1

Truncation Error (2)



Using HA – 3 logic gates per bit

Word-Serial: minimal extra hardware

Word-Parallel: 2 half adders per stage → Loop Unrolled Arch

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