Hybrid CORDIC 1.A Sine/Cosine Generator Algorithms

20171104

.

The details moved to

https://en.wikiversity.org/wiki/Butterfly_Hardware_Implementations

.

Wilson ROM based Sine/Cosine Generation

[24] Fu & Willson Sine / Cosine Generation

ROM-based

For high resolution, ROM size grows exponentially

quater-wave symmetry

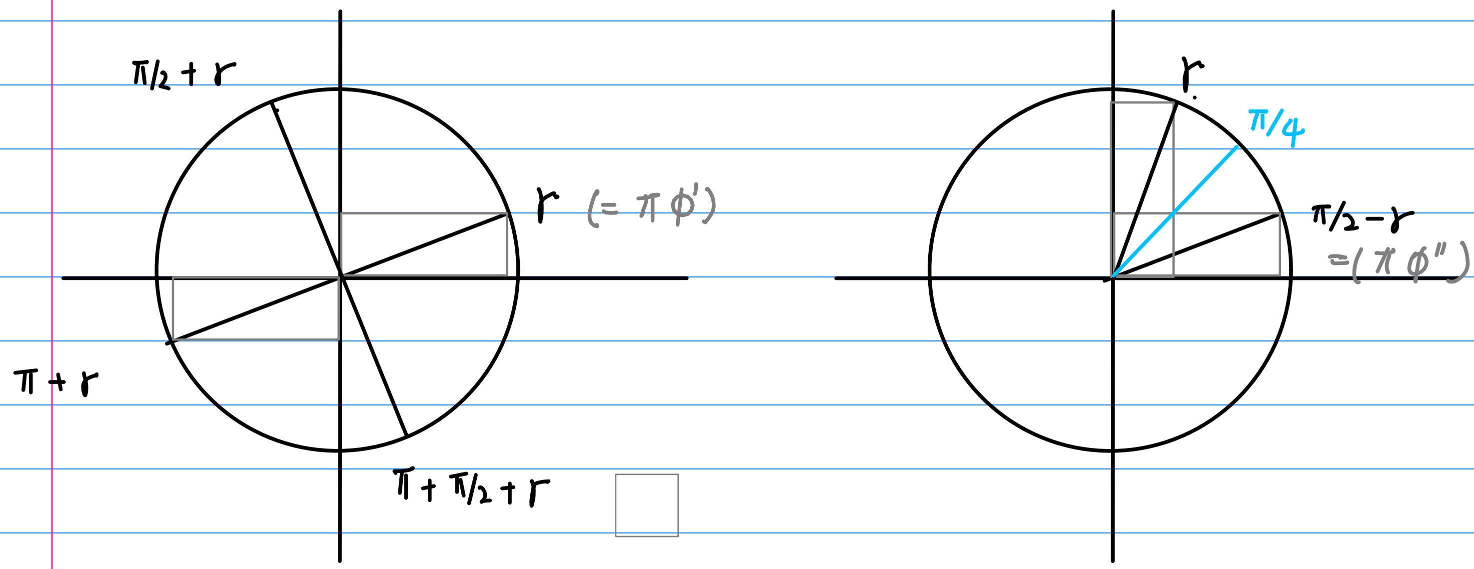
$$\sin\theta = \cos\left(\frac{\pi}{2} - \theta\right)$$

$$\phi \quad [0, 2\pi] \longrightarrow [0, \frac{\pi}{4}]$$

conditionally interchanging inputs X_0 & Y_0
Conditionally interchanging and negating outputs X & Y

$$X = X_0 \cos\phi - Y_0 \sin\phi$$
$$Y = Y_0 \cos\phi + X_0 \sin\phi$$

Madisetti VLSI arch



for frequency synthesis

argument: signed normalized by π angle $[-1, 1]$

binary representation of a radian angle required

$[-1, 1] \longrightarrow [0, \pi/4] \longrightarrow$ sine/cosine generator

ϕ

$\theta = \pi\phi$

① a phase accumulator ϕ $[-1, 1]$
② a radian converter $\phi \to \theta$
③ a sine/cosine generator $\sin\theta$, $\cos\theta$
④ an output stage $\sin\theta$, $\cos\theta$

$\sin\theta \to \sin\pi\phi$, $\cos\theta \to \cos\pi\phi$

.

output stage $\qquad \sin\theta \to \sin\pi\phi \qquad [-\pi, +\pi]$

$\qquad\qquad\qquad \cos\theta \to \cos\pi\phi$

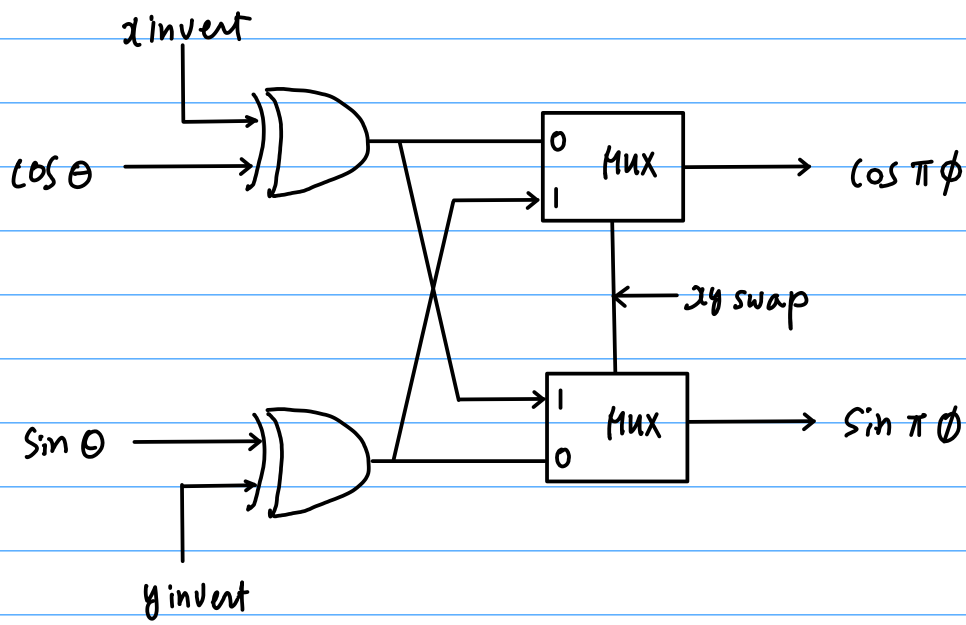
negation / interchange

MSB1, MSB2, MSB3 → x invert, y invert, xy swap

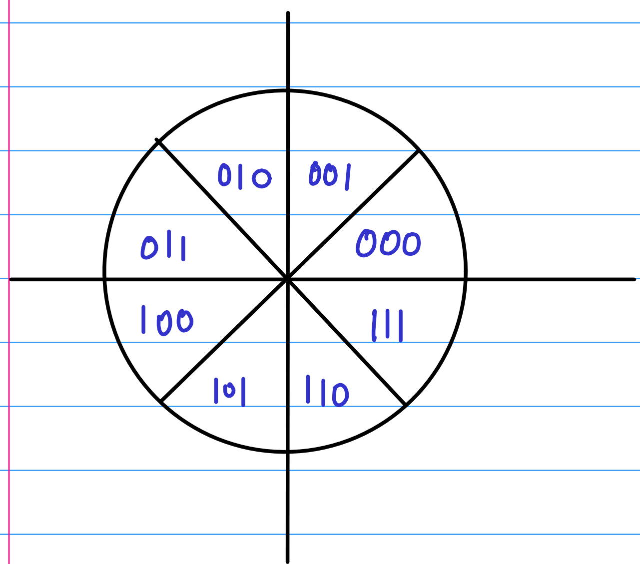
x invert, y invert: the negation of $\cos\theta = X_{N+1}$, $\sin\theta = Y_{N+1}$

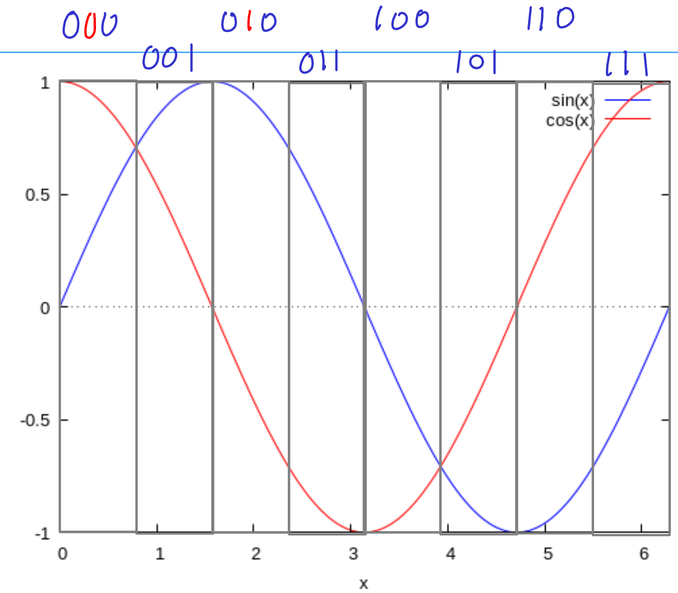
xy swap: Interchange

negate before swap



Madisetti & Willson, DDS Freq synthesizer





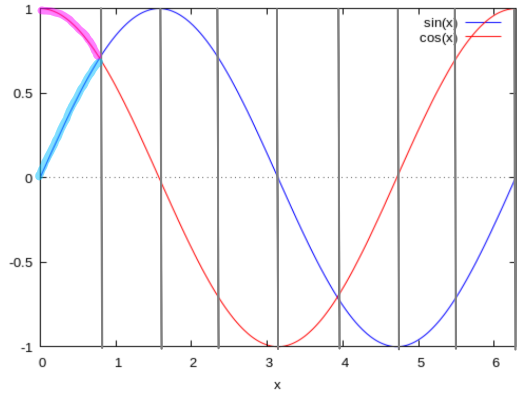
	cos	sin.			
	xinv	yinv	swap	$\cos \pi \theta$	$\sin \pi \phi$
0 0 0	0	0	0	$\cos\theta$	$\sin\theta$
0 0 1	0	0	1	$\sin\theta$	$\cos\theta$
0 1 0	0	1	1	$-\sin\theta$	$\cos\theta$
0 1 1	1	0	0	$-\cos\theta$	$\sin\theta$
1 0 0	1	1	0	$-\cos\theta$	$-\sin\theta$
1 0 1	1	1	1	$-\sin\theta$	$-\cos\theta$
1 1 0	1	0	1	$\sin\theta$	$-\cos\theta$
1 1 1	0	1	0	$\cos\theta$	$-\sin\theta$

Madisetti & Willson, DDS Freq synthesizer

0

$\cos\theta$

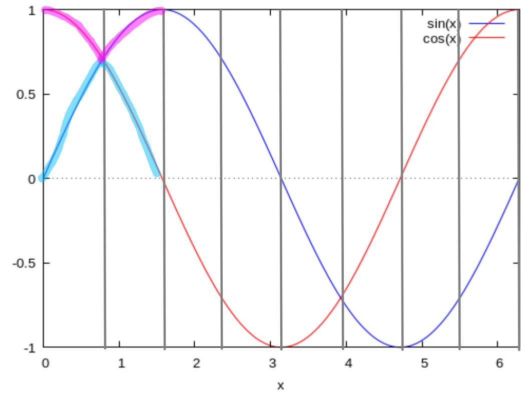
$\sin\theta$



1

$\sin\theta$

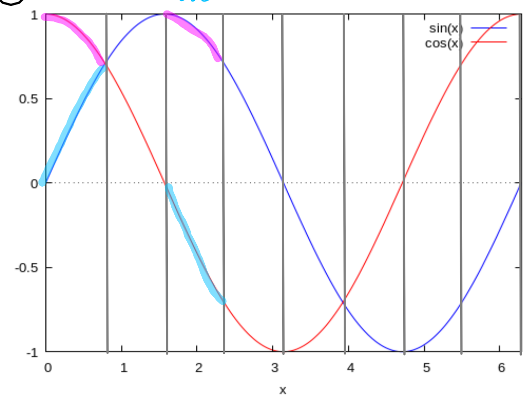
$\cos\theta$



2

$-\sin\theta$

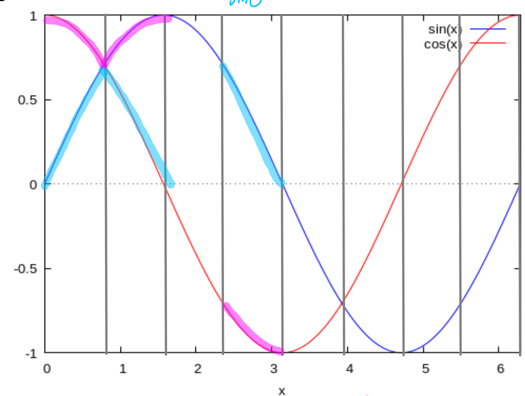
$\cos\theta$



3

$-\cos\theta$

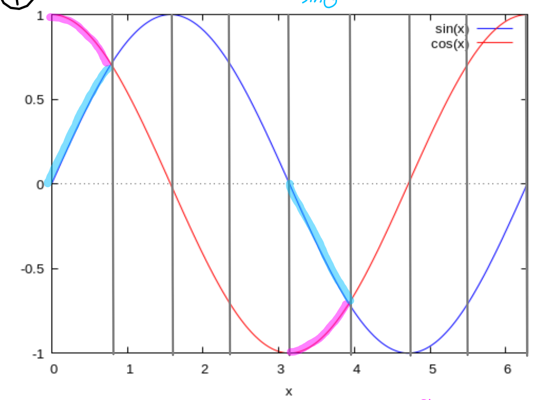
$\sin\theta$



4

$-\cos\theta$

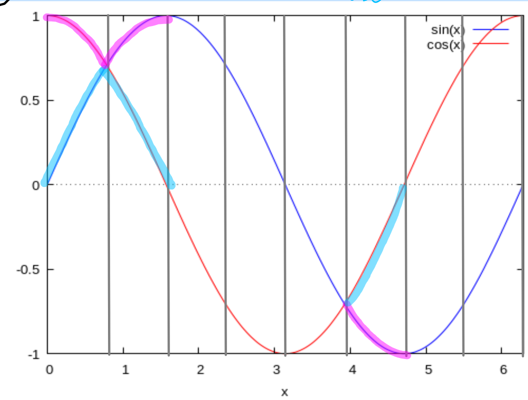
$-\sin\theta$



5

$-\sin\theta$

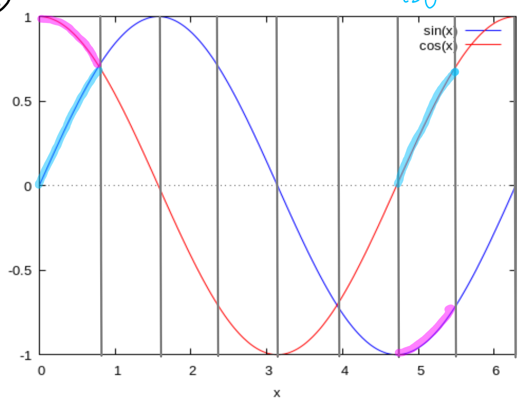
$-\cos\theta$



6

$\sin\theta$

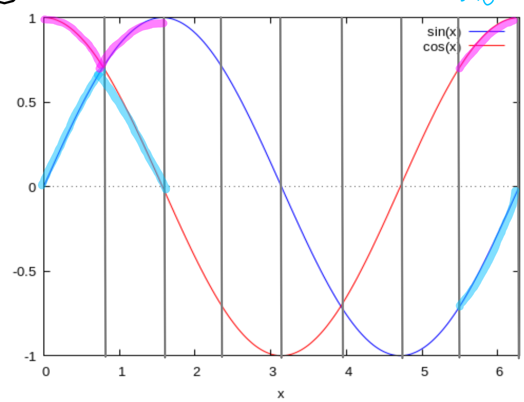
$-\cos\theta$



7

$\cos\theta$

$-\sin\theta$

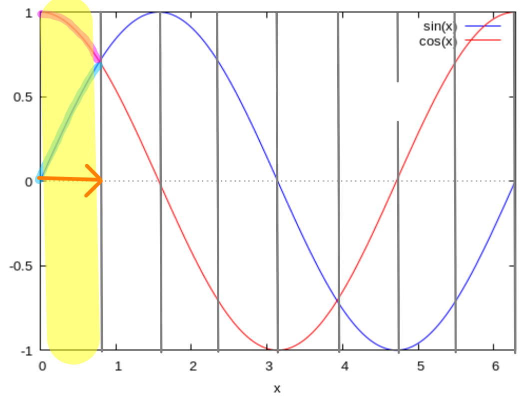


Madisetti & Willson, DDS Freq synthesizer

$$\begin{cases} \cos\phi \\ \sin\phi \end{cases}$$

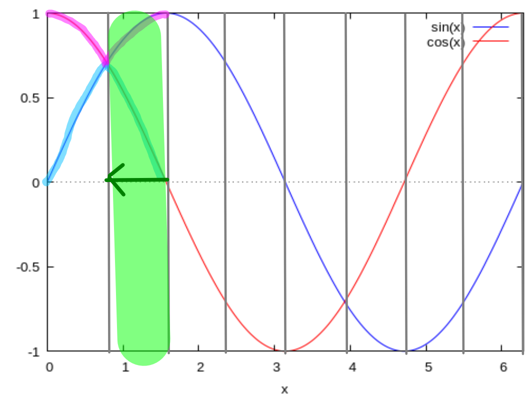
0

$\cos\theta$
$\sin\theta$



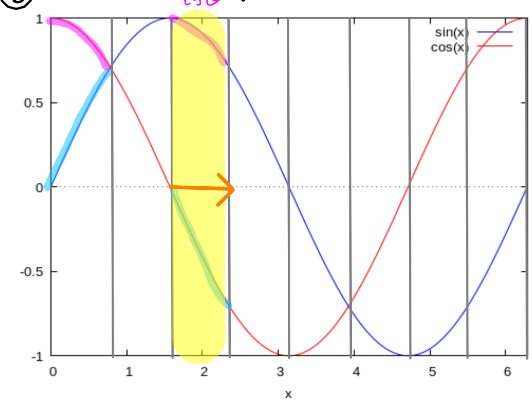
1

$\sin\theta$
$\cos\theta$



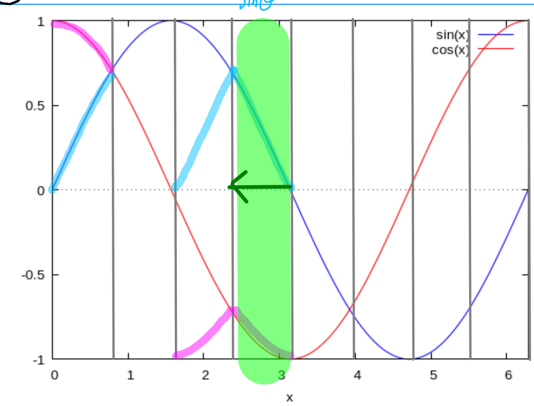
2

$-\sin\theta$
$\cos\theta$



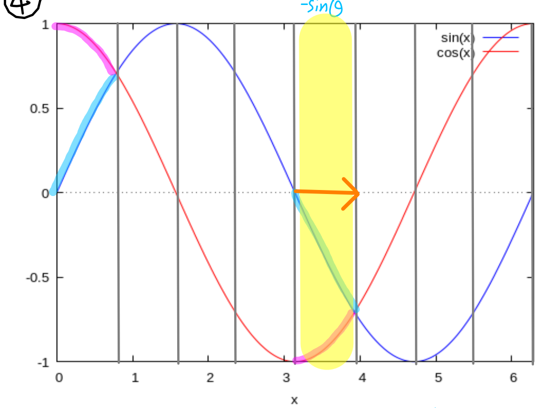
3

$-\cos\theta$
$\sin\theta$



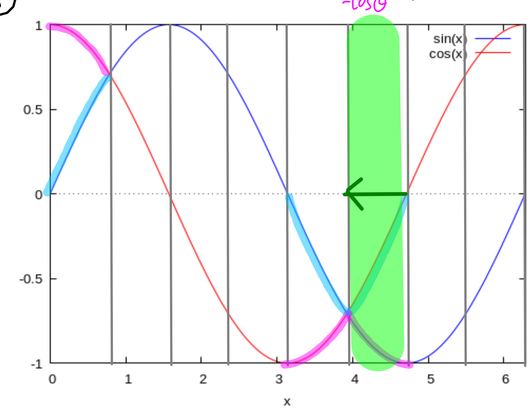
4

$-\cos\theta$
$-\sin\theta$



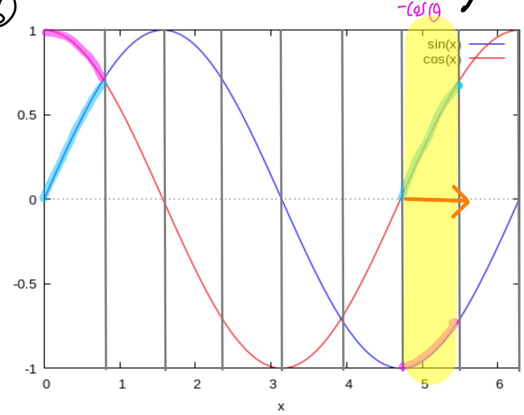
5

$-\sin\theta$
$-\cos\theta$



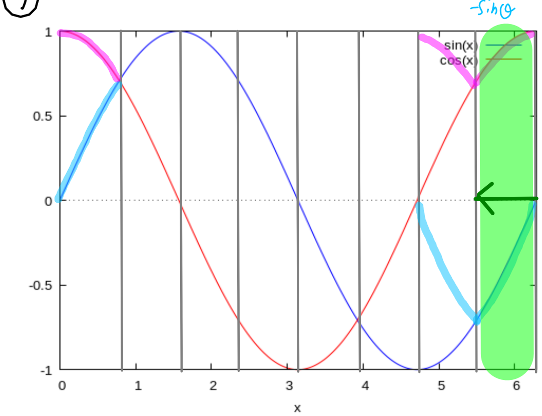
6

$\sin\theta$
$-\cos\theta$



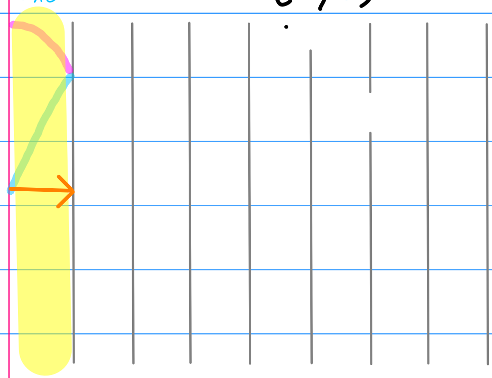
7

$\cos\theta$
$-\sin\theta$

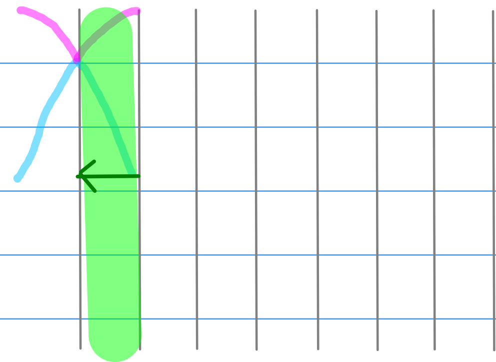


$\sin\phi$

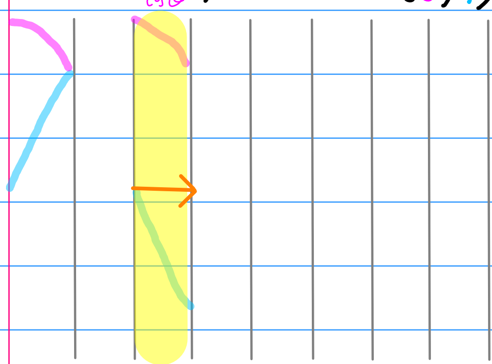
0 $\cos\theta$ / $\sin\theta$ $+ +$ $(0, 0)$



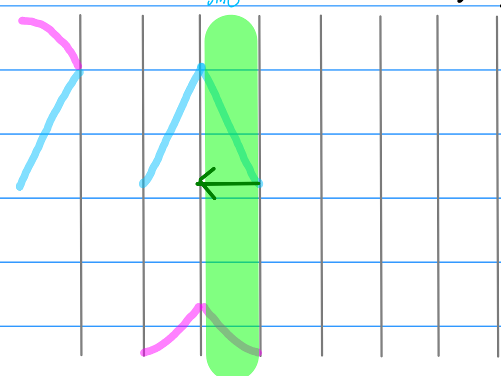
1 $\sin\theta$ / $\cos\theta$) $+ +$ $(0, 0)$



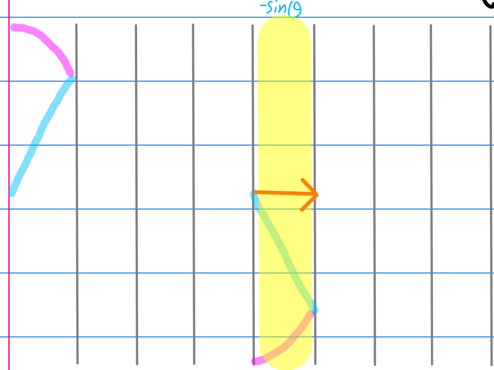
2 $-\sin\theta$ / $\cos\theta$) $+ -$ $(0, 1)$



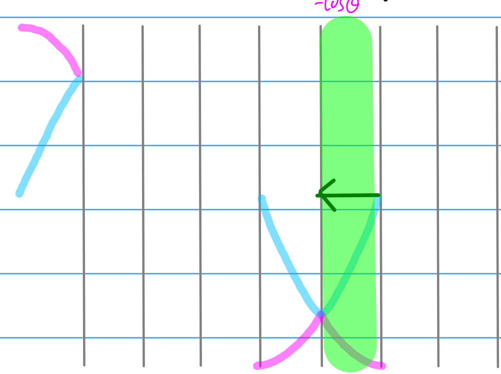
3 $-\cos\theta$ / $\sin\theta$ $- +$ $(1, 0)$



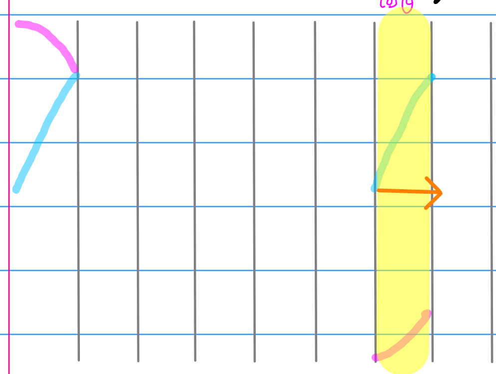
4 $-\cos\theta$ / $-\sin\theta$ $- -$ $(1, 1)$



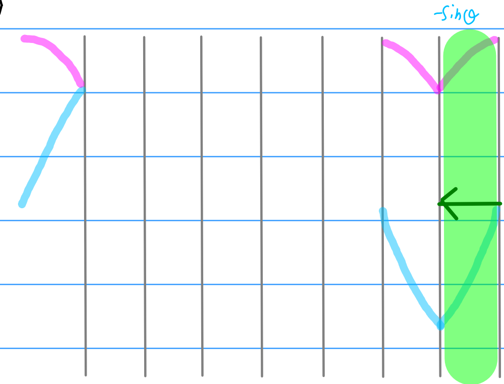
5 $-\sin\theta$ / $-\cos\theta$) $- -$ $(1, 1)$



6 $\sin\theta$ / $-\cos\theta$) $- +$ $(1, 0)$



7 $\cos\theta$ / $-\sin\theta$ $+ -$ $(0, 1)$



Madisetti & Willson, DDS Freq synthesizer

	xinv	yinv	swap	$\cos \pi \phi$	$\sin \pi \phi$
0 0 0	0	0	0	$\cos \theta$	$\sin \theta$
0 0 1	0	0	1	$\sin \theta$	$\cos \theta$
0 1 0	0	1	1	$-\sin \theta$	$\cos \theta$
0 1 1	1	0	0	$-\cos \theta$	$\sin \theta$
1 0 0	1	1	0	$-\cos \theta$	$-\sin \theta$
1 0 1	1	1	1	$-\sin \theta$	$-\cos \theta$
1 1 0	1	0	1	$\sin \theta$	$-\cos \theta$
1 1 1	0	1	0	$\cos \theta$	$-\sin \theta$

0 0	0 0
0 1	1 0
1 1	1 1
1 0	0 1

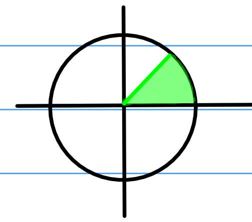
0	0	0	0
0	1	1	0
1	1	1	1
1	0	0	1

Angle Recoding

given angle θ (in radian)

$0 \leq \theta \leq \pi/4 < 1$

$\pi/4 \Rightarrow 0.785398163$



$$\theta = \sum_{k=1}^{N} b_k \theta_k$$

Binary Representation

$b_k \in \{0, 1\}$

$\theta_k = 2^{-k}$

(N+1) bit fractional binary

Sign + N bit $\Rightarrow$ | S | b_1 b_2 .... b_N |

assume θ is positive $\boxed{b_0 = 0}$ $\boxed{S = 0}$

$$\boxed{\theta = \sum_{k=1}^{N} b_k 2^{-k} = \phi_0 + \sum_{k=2}^{N+1} r_k 2^{-k}}$$

$r_k \in \{-1, +1\}$ signed digits

ϕ_0 constant

⊕ subrotation by 2^{-k}

2 equal ⊕ half rotations by 2^{-k-1}

⓪ subrotation

2 equal opposite half rotations by $\pm 2^{-k-1}$

Binary Representation

$b_k = 1$: rotation by 2^{-k}

$b_k = 0$: zero rotation

k-th rotation

- fixed rotation by 2^{-k-1}
- pos rotation ← $b_k = 1$
  neg rotation ← $b_k = 0$

Combining all the fixed rotations

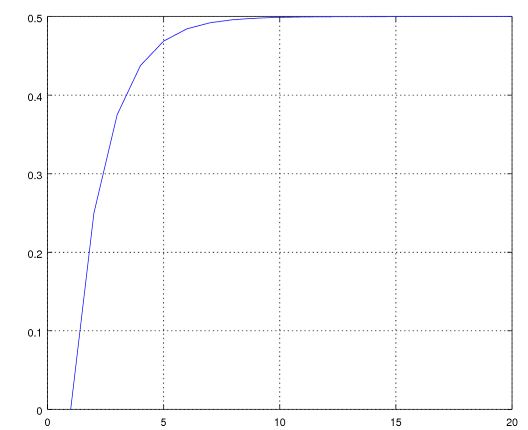
→ initial fixed rotation ϕ_0

	b_1 2^{-1}	b_2 2^{-2}	b_3 2^{-3}		b_N 2^{-N}
fixed $\Rightarrow$	$+2^{-2}$	$+2^{-3}$	$+2^{-4}$		$+2^{-N-1}$
	$(b_1=1)$ $+2^{-2}$ $(b_1=0)$ -2^{-2}	$(b_2=1)$ $+2^{-3}$ $(b_2=0)$ -2^{-3}	$(b_3=1)$ $+2^{-4}$ $(b_3=0)$ -2^{-4}		$(b_N=1)$ $+2^{-N-1}$ $(b_N=0)$ -2^{-N-1}

initial fixed rotation

$$\phi_0 = \frac{1}{2^2} + \frac{1}{2^3} + \cdots + \frac{1}{2^{N+1}}$$

$$= \frac{\frac{1}{2^2}\left(1 - \frac{1}{2^N}\right)}{\left(1 - \frac{1}{2}\right)} = \frac{1}{2}\left(1 - \frac{1}{2^N}\right) = \frac{1}{2} - \frac{1}{2^{N+1}}$$



Signed Digit Recoding

the rotation after recoding

— a fixed initial rotation ϕ_0

a sequence of ⊕/⊖ rotations

$b_k = 1 \quad + 2^{-k-1}$ rotation

$b_k = 0 \quad - 2^{-k-1}$ rotation

$r_k = (2b_{k-1} - 1)$

$2 \cdot 1 - 1 = +1 \qquad b_{k-1} = 1 \longrightarrow r_k = +1$

$2 \cdot 0 - 1 = -1 \qquad b_{k-1} = 0 \longrightarrow r_k = -1$

The recoding need not be explicitly performed

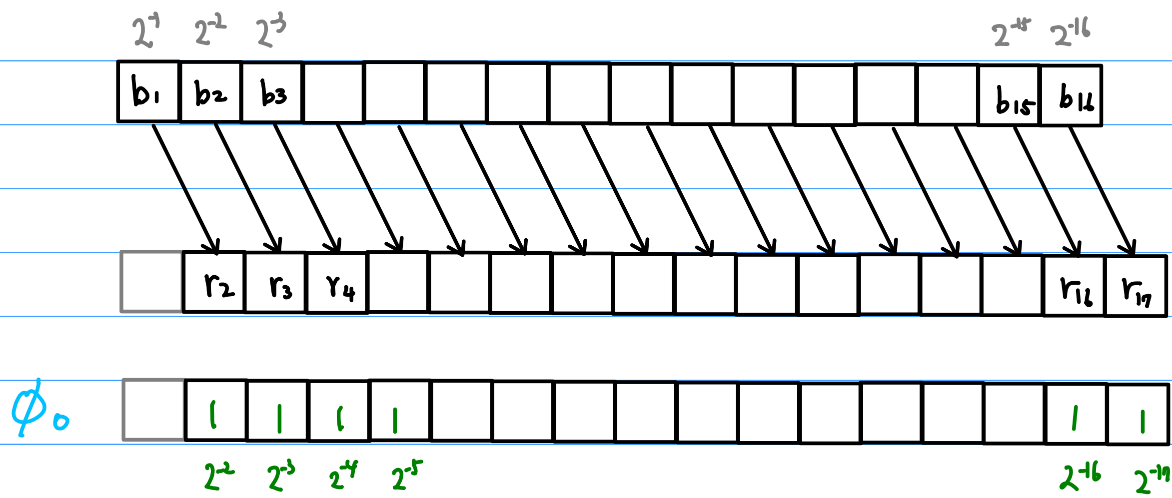
Simply replacing $b_k = 0$ with -1

This recoding maintains

a constant scaling factor K

$$\theta = \sum_{k=1}^{N} b_k 2^{-k} = \phi_0 + \sum_{k=2}^{N+1} r_k 2^{-k}$$

Binary Representation $\{b_k\}$



Signed Digit Recoding $\{r_k\}$

MSB_1	MSB_2	MSB_3

$\longrightarrow$ $0 < \theta'' < 1$ $\longrightarrow$ recoding $\{r_k\}$

$$\boxed{\sum_{k=1}^{N} b_k \cdot 2^{-k} = \phi_0 + \sum_{k=2}^{N+1} r_k \cdot 2^{-k}}$$

$b_k \in \{0, 1\}$ $\qquad$ $r_k \in \{-1, +1\}$

$$\begin{cases} b_k = 1 \longrightarrow r_{k+1} = +1 \\ b_k = 0 \longrightarrow r_{k+1} = -1 \end{cases} \qquad \boxed{r_k = (2b_{k-1} - 1)}$$

ϕ_0 depends only on bit width N

for fixed N, $\boxed{\phi_0 = \frac{1}{2} - \frac{1}{2^{N+1}}}$ is a constant

The Scaling K.

The initial rotation $\phi_0 = \frac{1}{2} - \frac{1}{2^{N+1}}$

rotation Starting point

$$(X_0, Y_0) = (K\cos\phi_0, \ K\sin\phi_0)$$

rotation always starts from this fixed point.

Cascade of feed forward rotational stages

$$\theta \rightarrow \boxed{MSB_1 \mid MSB_2 \mid MSB_3} \rightarrow \theta'' \rightarrow b_k 2^{-k} \rightarrow r_k 2^{-k}$$

binary representation　　　recoding

$\left\{ \begin{array}{l} \text{no comparison} \\ \text{no error build up} \end{array} \right.$

① $\boxed{\theta_k = \tan^{-1} 2^{-k}}$ traditional CORDIC

② $\boxed{\theta_k = 2^{-k}}$ possible because $\theta'' < 1$

$$\begin{bmatrix} 1 & -\sigma_k \tan(2^{-k}) \\ \sigma_k \tan(2^{-k}) & 1 \end{bmatrix}$$

① $\boxed{\theta_k = \tan^{-1} 2^{-k}}$ traditional CORDIC

$$\begin{bmatrix} 1 & -\sigma_k \tan(\tan^{-1} 2^{-k}) \\ \sigma_k \tan(\tan^{-1} 2^{-k}) & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & -\sigma_k 2^{-k} \\ \sigma_k 2^{-k} & 1 \end{bmatrix}$$

☆ ② $\boxed{\theta_k = 2^{-k}}$ possible because $\theta'' < 1$

$$\begin{bmatrix} 1 & -\sigma_k \tan(2^{-k}) \\ \sigma_k \tan(2^{-k}) & 1 \end{bmatrix} \rightarrow \left\{ \begin{array}{l} \begin{bmatrix} 1 & -\sigma_k \tan(2^{-k}) \\ \sigma_k \tan(2^{-k}) & 1 \end{bmatrix} \\ \\ \begin{bmatrix} 1 & -\sigma_k 2^{-k} \\ \sigma_k 2^{-k} & 1 \end{bmatrix} \end{array} \right.$$

$\tan(2^{-k}) \cong 2^{-k} \quad k \geq k_0$

$$K = \cos(\theta_0)\cos(\theta_1)\cdots\cos(\theta_N)$$

$$K \begin{bmatrix} 1 & -\tan(\sigma_0\theta_0) \\ \tan(\sigma_0\theta_0) & 1 \end{bmatrix} \begin{bmatrix} 1 & -\tan(\sigma_1\theta_1) \\ \tan(\sigma_1\theta_1) & 1 \end{bmatrix} \cdots \begin{bmatrix} 1 & -\tan(\sigma_N\theta_N) \\ \tan(\sigma_N\theta_N) & 1 \end{bmatrix}$$

$$\boxed{\theta_k = \tan^{-1} 2^{-k}} \rightarrow \tan\theta_k = 2^{-k}$$

$$K \begin{bmatrix} 1 & -\sigma_0 2^{-0} \\ \sigma_0 2^{-0} & 1 \end{bmatrix} \begin{bmatrix} 1 & -\sigma_1 2^{-1} \\ \sigma_1 2^{-1} & 1 \end{bmatrix} \cdots \begin{bmatrix} 1 & -\sigma_N 2^{-N} \\ \sigma_N 2^{-N} & 1 \end{bmatrix}$$

↳ shift-and-add

$$K = \cos(\theta_0)\cos(\theta_1)\cdots\cos(\theta_N)$$

$$K \begin{bmatrix} 1 & -\tan(\sigma_0\theta_0) \\ \tan(\sigma_0\theta_0) & 1 \end{bmatrix} \begin{bmatrix} 1 & -\tan(\sigma_1\theta_1) \\ \tan(\sigma_1\theta_1) & 1 \end{bmatrix} \cdots \begin{bmatrix} 1 & -\tan(\sigma_N\theta_N) \\ \tan(\sigma_N\theta_N) & 1 \end{bmatrix}$$

$$\star \quad \boxed{\theta_k = 2^{-k}} \longrightarrow \tan\theta_k = \tan 2^{-k}$$

$$K \begin{bmatrix} 1 & -\sigma_0\tan(2^{-0}) \\ \sigma_0\tan(2^{-0}) & 1 \end{bmatrix} \begin{bmatrix} 1 & -\sigma_1\tan(2^{-1}) \\ \sigma_1\tan(2^{-1}) & 1 \end{bmatrix} \cdots \begin{bmatrix} 1 & -\sigma_N\tan(2^{-N}) \\ \sigma_N\tan(2^{-N}) & 1 \end{bmatrix}$$

$$\tan(2^{-k}) \cong 2^{-k} \qquad k \geq k_0$$

$$K \begin{bmatrix} 1 & -\sigma_0\tan(2^{-0}) \\ \sigma_0\tan(2^{-0}) & 1 \end{bmatrix} \cdots \begin{bmatrix} 1 & -\sigma_{k_0} 2^{-k_0} \\ \sigma_{k_0} 2^{-k_0} & 1 \end{bmatrix} \cdots \begin{bmatrix} 1 & -\sigma_N 2^{-N} \\ \sigma_N 2^{-N} & 1 \end{bmatrix}$$

simple shift-and-add

the $\tan\theta_k$ multipliers used in the first few subrotation stages cannot be implemented as simple shift-and-add operations the subangles $\boxed{\theta_k = 2^{-k}}$ used in recoding

the subangles $\boxed{\theta_k = \tan^{-1}(2^{-k})}$ used in CORDIC

$\tan \theta_k$ multipliers used
in the first few subrotation stages
cannot be implemented
as a simple shift-and-add operations

→ ROM implementation

reduced chip area
higher operating speed.

the rotations always start from the fixed point

the computational sequence

- a cascade of feed forward rotational stages

the desired output precision in bits
determines the number of stages

by always starting the sequence of rotations
from a fixed point,
the algorithm does not suffer from an error build up

which limits the accuracy of most recursive
digital oscillator structures

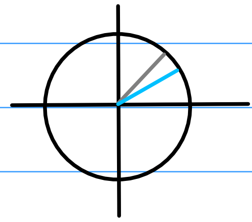
Architecture

① phase accumulator $\phi \in [-1, +1]$

② radian converter $\phi \to \theta \in [0, \frac{\pi}{4}]$

③ Sine/cosine generator $\sin(\theta)$ $\cos(\theta)$

④ Output stage $\sin(\pi\phi)$ $\cos(\pi\phi)$



$\phi \in [-1, +1]$ normalized angle

$\phi \in [-\pi, +\pi] \to \theta \in [0, \frac{\pi}{4}]$ 1st half quadrant

$\sin(\theta)$ $\cos(\theta)$

$\sin(\pi\phi)$ $\cos(\pi\phi)$

Overflowing 2's complement accumulator

normalized by π angle ϕ

need radian angle $\theta \in [0, \frac{\pi}{4}]$

$0 < \theta < 1$ rad

N-bit binary representation of θ

controls the direction of subrotation

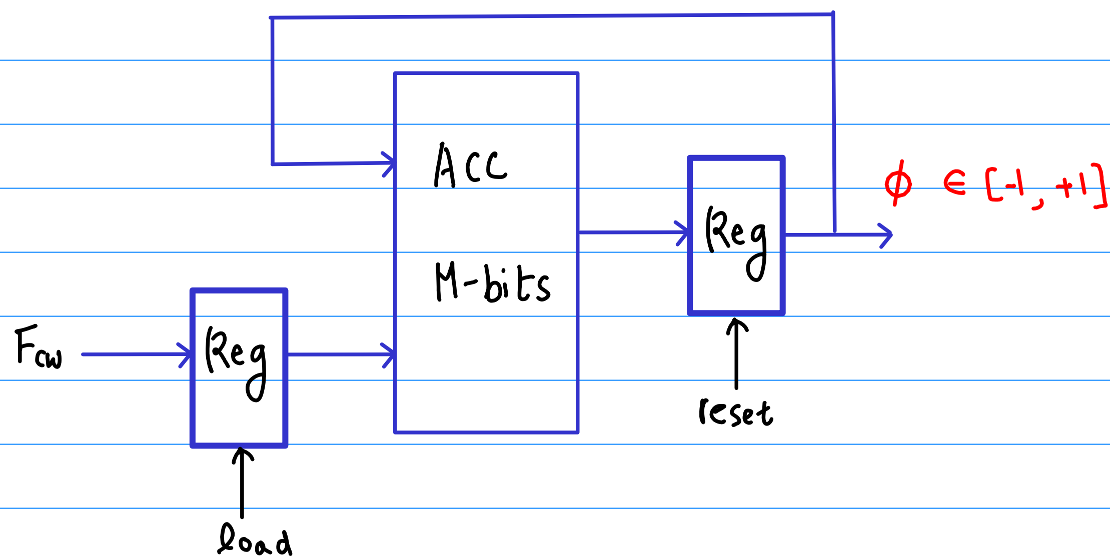
N-bit precision of $\cos\theta$ & $\sin\theta$

Output stage $\theta \to \pi\phi$

$\sin\theta \to \sin\pi\phi$

$\cos\theta \to \cos\pi\phi$

① phase accumulator



M-bit adder — repeatedly increments the phase a

by F_{cw} at each clock cycle

frequency control word

at time n, $\boxed{\phi = n F_{cw} / 2^M}$

$$\cos\phi = \cos(n F_{cw}/2^M)$$
$$\sin\phi = \sin(n F_{cw}/2^M)$$

$$-1 < \phi < +1$$

normalized angle

Normalized Angle

at time n,

$$\boxed{\phi = n F_{cw} / 2^M}$$

$$\cos\phi = \cos(n F_{cw} / 2^M)$$
$$\sin\phi = \sin(n F_{cw} / 2^M)$$

$-1 < \phi < +1$ $\quad\quad$ ϕ radians$/\pi$

$-\pi < \pi\phi < +\pi$ $\quad\quad$ $\pi\phi$ radians

$-\pi < \theta < +\pi$ $\quad\quad$ θ radians

$$\theta = \pi\phi$$

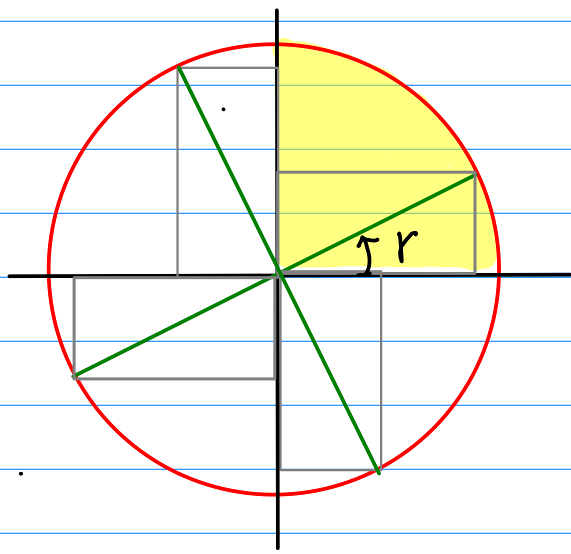
$$\theta \in [-\pi, +\pi]$$
$$\phi \in [-1, +1]$$

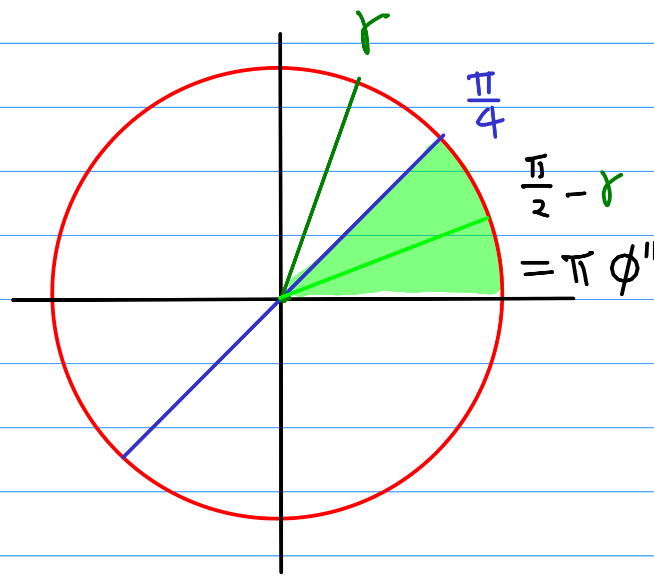
② Radian Converter

$$\theta = \pi\phi \rightarrow \theta' = \pi\phi' \rightarrow \theta'' = \pi\phi''$$

$$\theta \in [-\pi, +\pi] \rightarrow \theta' = [0, \pi/2] \rightarrow \theta'' = [0, \pi/4]$$

$$\phi \in [-1, +1] \rightarrow \phi' = [0, 0.5] \rightarrow \phi'' = [0, 0.25]$$





$$\theta' = \pi\phi' \qquad \theta'' = \pi\phi''$$

MSB$_1$
MSB$_2$
MSB$_3$

$$-1 < \phi < +1$$

$$-\pi < \pi\phi < +\pi$$

$$-\pi < \theta < +\pi$$

radian

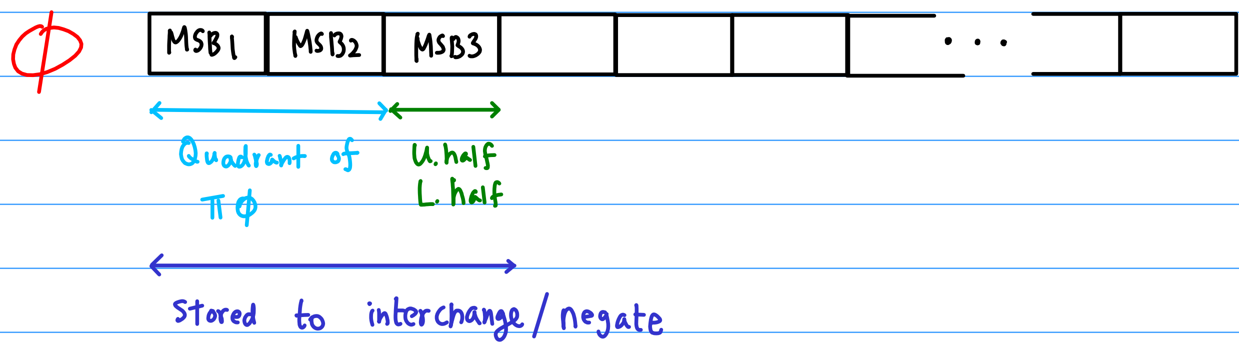
$$0 < \phi'' < 1/4$$

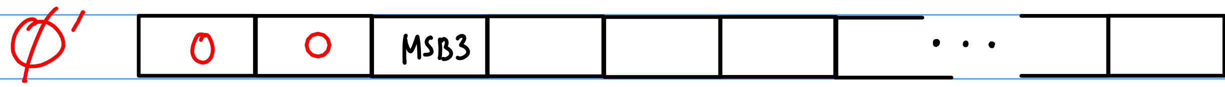
$$0 < \pi\phi'' < \pi/4$$

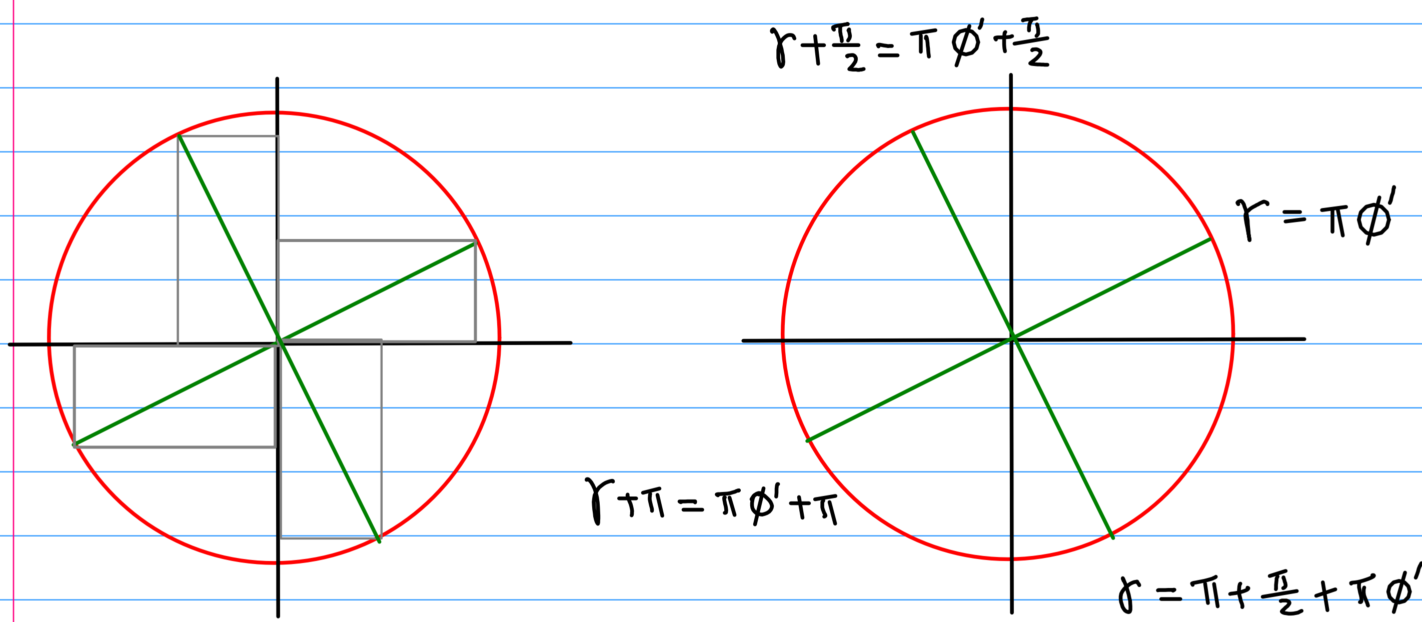
$$0 < \theta'' < 1$$

radian

normalized angle ϕ





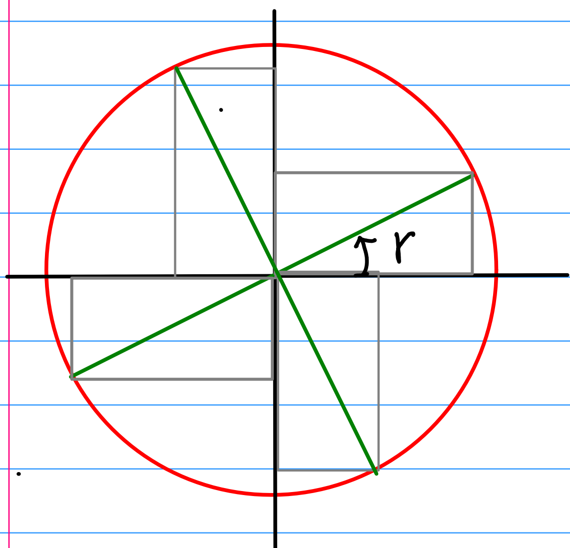


$\phi \rightarrow \phi' \rightarrow$

$\pi\phi'$	$+\ 0 \cdot \frac{\pi}{2}$	00
$\pi\phi'$	$+\ 1 \cdot \frac{\pi}{2}$	01
$\pi\phi'$	$+\ 2 \cdot \frac{\pi}{2}$	10
$\pi\phi'$	$+\ 3 \cdot \frac{\pi}{2}$	11

1st Quad

Quadrant Symmetry



$\boxed{0\,1}$ $\gamma + \frac{\pi}{2} = \pi\phi' + \frac{\pi}{2}$	$\boxed{0\,0}$ $\gamma = \pi\phi'$
$\boxed{1\,0}$ $\gamma + \pi = \pi\phi' + \pi$	$\boxed{1\,1}$ $\gamma + \frac{3\pi}{2} = \pi\phi' + \frac{3\pi}{2}$

0	1	+1
0	0	0
1	1	-1
1	0	-2

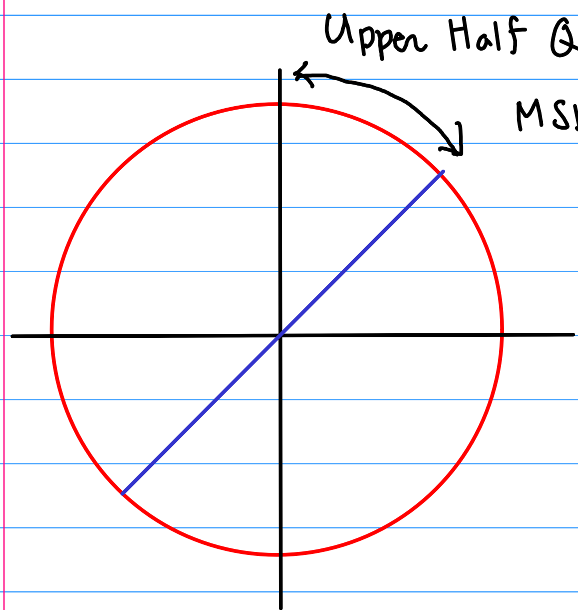
$$\theta = \pi\phi \Rightarrow \theta' = \pi\phi'$$

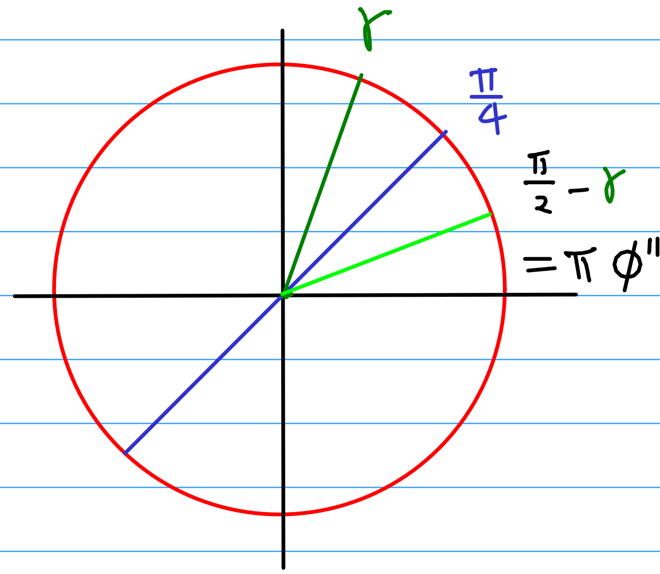
$$\theta \in [-\pi, +\pi] \Rightarrow \theta = [0, \tfrac{\pi}{2}]$$

$$\phi \in [-1, +1] \Rightarrow \phi' = [0, 0.5]$$

ϕ'									
	0	0	MSB3				...		

1st Quadrant



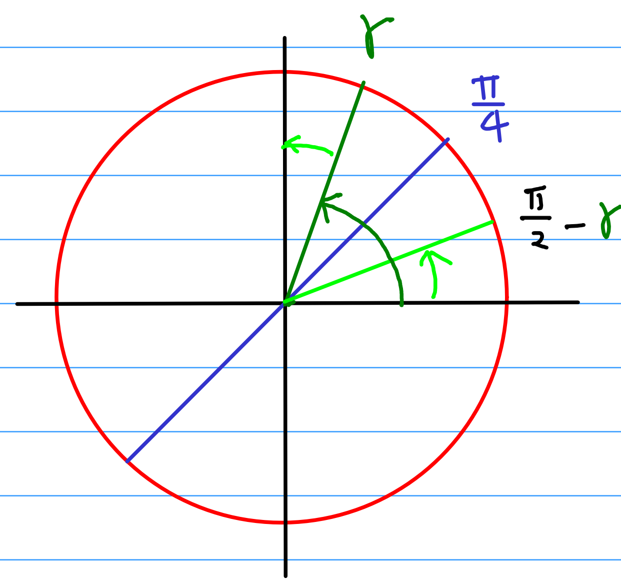


$\gamma > \frac{\pi}{4}$: Upper Half $(MSB_3 = 1)$

$\gamma < \frac{\pi}{4}$: Lower Half $(MSB_3 = 0)$

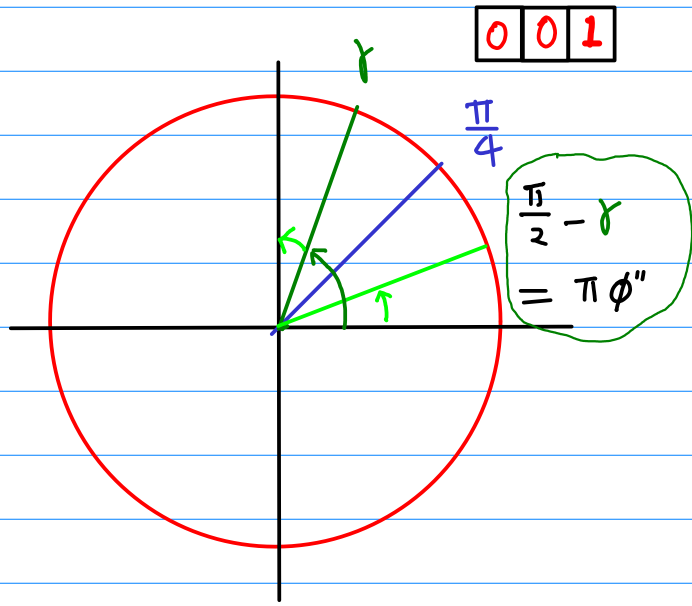
$$\cos\gamma = \sin\left(\frac{\pi}{2} - \gamma\right)$$

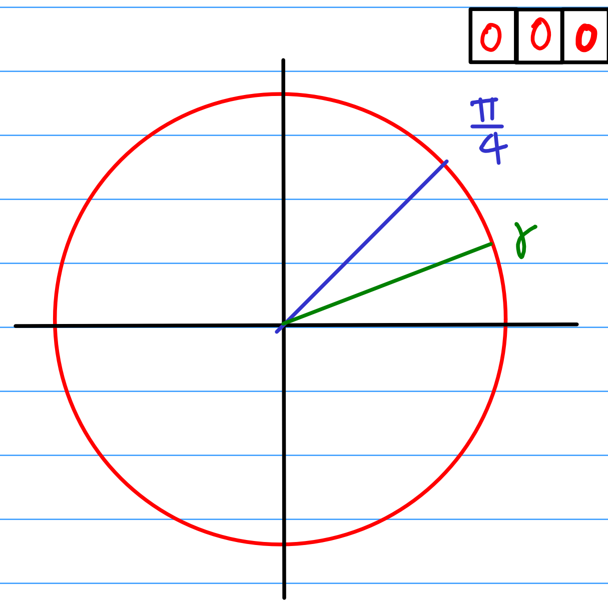
$$\sin\gamma = \cos\left(\frac{\pi}{2} - \gamma\right)$$



$\gamma > \frac{\pi}{4}$ $\quad$ $\frac{\pi}{2} - \gamma < \frac{\pi}{4}.$

$\pi/4$ mirror





$\frac{\pi}{4} < \gamma < \frac{\pi}{2}$

$\phi'' = 0.5 - \phi'$

$0 < \gamma < \frac{\pi}{4}$

$\phi'' = \phi'$

$$\theta = \pi\phi \rightarrow \theta' = \pi\phi' \rightarrow \theta'' = \pi\phi''$$

$$\theta \in [-\pi, +\pi] \rightarrow \theta' = [0, \pi/2] \rightarrow \theta'' = [0, \pi/4]$$

$$\phi \in [-1, +1] \rightarrow \phi' = [0, 0.5] \rightarrow \phi'' = [0, 0.25]$$

ϕ'									
	0	0	MSB3					...	

MSB3 = 1 $\qquad \phi' > \frac{\pi}{4}$

$$\phi'' = \frac{\pi}{2} - \phi'$$

0.5	1	0	0					...	
$\ominus\ \phi'$	0	0	MSB3					...	

$$\begin{cases} MSB_3 = 0 & \phi'' = \phi' \\ MSB_3 = 1 & \phi'' = 0.5 - \phi' \end{cases}$$

$\theta = \pi \phi''$ (Handwired Multiplier)

$0 < \theta < \frac{\pi}{4}$

$$\phi \longrightarrow \phi' \longrightarrow \phi''$$

1st Quad $\qquad$ Lower Half

$$\theta = \pi\phi \longrightarrow \theta' = \pi\phi' \longrightarrow \theta'' = \pi\phi''$$

$$\theta \in [-\pi, +\pi] \longrightarrow \theta' = [0, \pi/2] \longrightarrow \theta'' = [0, \pi/4]$$

$$\phi \in [-1, +1] \longrightarrow \phi' = [0, 0.5] \longrightarrow \phi'' = [0, 0.25]$$

$-\pi < \theta < +\pi$

$0 < \theta'' < 1$

$$\boxed{\begin{array}{l}\phi \in [-1, +1] \\ \theta \in [-\pi, +\pi]\end{array}} \xrightarrow[\boxed{MSB_2}]{\boxed{MSB_1}} \boxed{\begin{array}{l}\phi' = [0, 0.5] \\ \theta' = [0, \pi/2]\end{array}} \xrightarrow{\boxed{MSB_3}} \boxed{\begin{array}{l}\phi'' = [0, 0.25] \\ \theta'' = [0, \pi/4]\end{array}}$$

$\boxed{\text{Compute}}$

$$\boxed{\begin{array}{l}\phi \in [-1, +1] \\ \theta \in [-\pi, +\pi]\end{array}} \xleftarrow{\begin{array}{c}\boxed{MSB_1} \\ \boxed{MSB_2}\end{array}} \boxed{\begin{array}{l}\phi' = [0, 0.5] \\ \theta' = [0, \pi/2]\end{array}} \xleftarrow{\boxed{MSB_3}} \boxed{\begin{array}{l}\phi'' = [0, 0.25] \\ \theta'' = [0, \pi/4]\end{array}}$$

$-\pi < \theta < +\pi$

$0 < \theta'' < 1$

recoding is possible

$$\boxed{\theta''} = \pi \boxed{\phi''}$$

$0 < \theta'' < 1$

radian angle

$0 < \theta'' < \pi/4$

normalized angle

$0 < \phi'' < 0.25$

The multiplication by π

→ could have used a hardwired multiplier

→ but don't have to use a multiplier at all

① in table lookup DDFS architecture

→ here, the multiplication by π is implicit

② in CORDIC architecture

the elementary angle are divided by π

$$\theta_k = \tan^{-1}(2^{-k}) / 2\pi$$

the direction of subrotations are determined by the sign of angle difference

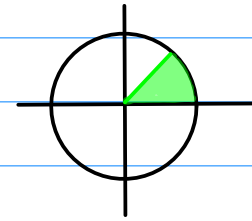
therefore the multiplication by π is <u>not</u> <u>necessary</u>

③ Sine / Cosine Generator

given angle θ (in radian)

$0 \leq \theta \leq \pi/4 < 1$

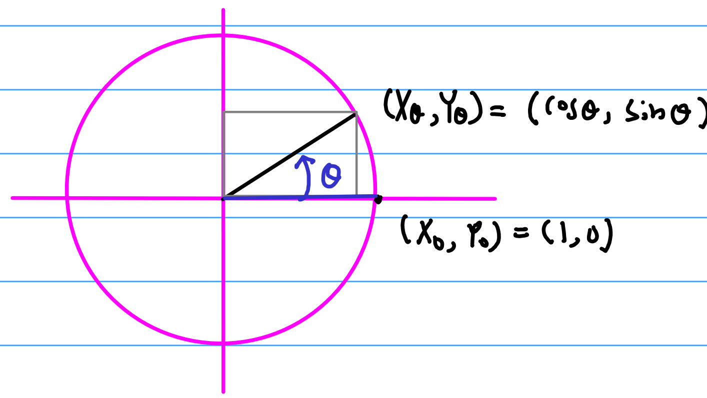
$\pi/4 \Downarrow 0.785398163$



compute $\boxed{\cos\theta}$, $\boxed{\sin\theta}$?

$$\begin{bmatrix} X_\theta \\ Y_\theta \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} X_0 \\ Y_0 \end{bmatrix}$$

$$= \cos\theta \begin{bmatrix} 1 & -\tan\theta \\ \tan\theta & 1 \end{bmatrix} \begin{bmatrix} X_0 \\ Y_0 \end{bmatrix}$$



$$\begin{bmatrix} X_\theta \\ Y_\theta \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} X_0 \\ Y_0 \end{bmatrix}$$

$$= \cos\theta \begin{bmatrix} 1 & -\tan\theta \\ \tan\theta & 1 \end{bmatrix} \begin{bmatrix} X_0 \\ Y_0 \end{bmatrix}$$

$$= \cos\theta \begin{bmatrix} 1 & -\tan\theta \\ \tan\theta & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

a **sequence** of **subrotations** of the priori **known** angle

Suppose: θ as a sequence of sub-rotation

$\{\theta_k\}$ the subrotation angles are <u>known</u> a priori

then $\theta = \sigma_0 \theta_0 + \sigma_1 \theta_1 + \cdots + \sigma_N \theta_N$

$\sigma_k = \{-1, 0, +1\}$

$$\theta = \sigma_0 \theta_0 + \sigma_1 \theta_1 + \cdots + \sigma_N \theta_N$$

$$\sigma_k = \{-1, 0, +1\}$$

$$\sigma_0 \theta_0 \rightarrow \cos(\sigma_0 \theta_0) \begin{bmatrix} 1 & -\tan(\sigma_0 \theta_0) \\ \tan(\sigma_0 \theta_0) & 1 \end{bmatrix}$$

$$\sigma_1 \theta_1 \rightarrow \cos(\sigma_1 \theta_1) \begin{bmatrix} 1 & -\tan(\sigma_1 \theta_1) \\ \tan(\sigma_1 \theta_1) & 1 \end{bmatrix}$$

$$\sigma_N \theta_N \rightarrow \cos(\sigma_N \theta_N) \begin{bmatrix} 1 & -\tan(\sigma_N \theta_N) \\ \tan(\sigma_N \theta_N) & 1 \end{bmatrix}$$

$$\theta = \sigma_0 \theta_0 + \sigma_1 \theta_1 + \cdots + \sigma_N \theta_N$$

$$\sigma_k = \{-1, 0, +1\}$$

$$\begin{bmatrix} X_\theta \\ Y_\theta \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} X_0 \\ Y_0 \end{bmatrix}$$

$$= \boxed{\cos(\sigma_0 \theta_0)} \begin{bmatrix} 1 & -\tan(\sigma_0 \theta_0) \\ \tan(\sigma_0 \theta_0) & 1 \end{bmatrix} \boxed{\cos(\sigma_1 \theta_1)} \begin{bmatrix} 1 & -\tan(\sigma_1 \theta_1) \\ \tan(\sigma_1 \theta_1) & 1 \end{bmatrix}$$

$$\cdots \boxed{\cos(\sigma_N \theta_N)} \begin{bmatrix} 1 & -\tan(\sigma_N \theta_N) \\ \tan(\sigma_N \theta_N) & 1 \end{bmatrix}$$

$$= K \begin{bmatrix} 1 & -\tan(\sigma_0 \theta_0) \\ \tan(\sigma_0 \theta_0) & 1 \end{bmatrix} \begin{bmatrix} 1 & -\tan(\sigma_1 \theta_1) \\ \tan(\sigma_1 \theta_1) & 1 \end{bmatrix} \cdots \begin{bmatrix} 1 & -\tan(\sigma_N \theta_N) \\ \tan(\sigma_N \theta_N) & 1 \end{bmatrix}$$

$$K = \boxed{\cos(\sigma_0 \theta_0)} \boxed{\cos(\sigma_1 \theta_1)} \cdots \boxed{\cos(\sigma_N \theta_N)}$$ scale factor

$\sigma_k = +1$ positive angle rotation

$\sigma_k = -1$ negative angle rotation

$$\theta = \sigma_0 \theta_0 + \sigma_1 \theta_1 + \cdots + \sigma_N \theta_N$$

$$\sigma_k = \{-1, 0, +1\}$$

CORDIC Algorithm

$$\boxed{\begin{aligned} \theta_k &= \tan^{-1} 2^{-k} \\ \tan \theta_k &= 2^{-k} \\ \tan(\sigma_k \theta_k) &= (\sigma_k 2^{-k}) \end{aligned}}$$

$$\sigma_k = \{-1, +1\}$$

$$K \begin{bmatrix} 1 & -\tan(\sigma_0\theta_0) \\ \tan(\sigma_0\theta_0) & 1 \end{bmatrix} \begin{bmatrix} 1 & -\tan(\sigma_1\theta_1) \\ \tan(\sigma_1\theta_1) & 1 \end{bmatrix} \cdots \begin{bmatrix} 1 & -\tan(\sigma_N\theta_N) \\ \tan(\sigma_N\theta_N) & 1 \end{bmatrix}$$

$$K = \cos(\sigma_0\theta_0)\cos(\sigma_1\theta_1)\cdots\cos(\sigma_N\theta_N)$$ scale factor

$$K \begin{bmatrix} 1 & -\sigma_0 2^{-0} \\ \sigma_0 2^{-0} & 1 \end{bmatrix} \begin{bmatrix} 1 & -\sigma_1 2^{-1} \\ \sigma_1 2^{-1} & 1 \end{bmatrix} \cdots \begin{bmatrix} 1 & -\sigma_N 2^{-N} \\ \sigma_N 2^{-N} & 1 \end{bmatrix}$$

↳ shift-and-add

$$K = \cos(\theta_0)\cos(\theta_1)\cdots\cos(\theta_N)$$

constant ← each rotation
+/− rotation
is actually performed

$$\begin{pmatrix} X_0 \\ Y_0 \end{pmatrix} = \begin{pmatrix} K \\ 0 \end{pmatrix}$$

$$\begin{bmatrix} X_\theta \\ Y_\theta \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} X_0 \\ Y_0 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & -\sigma_0 2^{-0} \\ \sigma_0 2^{-0} & 1 \end{bmatrix} \begin{bmatrix} 1 & -\sigma_1 2^{-1} \\ \sigma_1 2^{-1} & 1 \end{bmatrix} \cdots \begin{bmatrix} 1 & -\sigma_N 2^{-N} \\ \sigma_N 2^{-N} & 1 \end{bmatrix} \begin{bmatrix} K \\ 0 \end{bmatrix}$$

$$K = \cos(\theta_0)\cos(\theta_1)\cdots\cos(\theta_N)$$

σ_k determines pos / neg subrotation by an angle θ_k

σ_k values are determined iteratively by the successive approximation

at the k-th iteration

- if the current appraximation $>$ the input angle θ then subtract θ_k
- if the current appraximation $<$ the input angle θ then add θ_k

CORDIC HW $\qquad\qquad$ $\frac{1}{3}$ of the total HW

ⓐ computes σ_k
updates the current approximation by the angle θ_k

ⓑ performs the rotation by θ_k

$$\begin{pmatrix} \text{addition} \\ \text{comparison} \end{pmatrix}$$

redundant CSA $\qquad$ $\boxed{\text{addition}}$ $\quad \sum \theta_k$
eliminates the carry propagate delay
improves the throughput

the evaluation of each σ_k $\qquad$ $\boxed{\text{comparison}}$
requires the knowledge of the sign difference
between two angles

the sign detection in redundant arithmetic
non-trivial, bottleneck

$$\theta = \sigma_0\theta_0 + \sigma_1\theta_1 + \cdots + \sigma_N\theta_N$$

$$\sigma_k = \{-1, 0, +1\}$$

Recoding Algorithm

$$\boxed{\begin{aligned}\theta_k &= 2^{-k} \\ \tan\theta_k &= \tan 2^{-k} \\ \tan(\sigma_k\theta_k) &= \tan(\sigma_k 2^{-k})\end{aligned}}$$

$$\sigma_k = \{-1, +1\}$$

$$\begin{bmatrix} X_\theta \\ Y_\theta \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}\begin{bmatrix} X_0 \\ Y_0 \end{bmatrix} = \cos\theta \begin{bmatrix} 1 & -\tan\theta \\ \tan\theta & 1 \end{bmatrix}\begin{bmatrix} X_0 \\ Y_0 \end{bmatrix}$$

$$K \begin{bmatrix} 1 & -\tan(\sigma_0\theta_0) \\ \tan(\sigma_0\theta_0) & 1 \end{bmatrix}\begin{bmatrix} 1 & -\tan(\sigma_1\theta_1) \\ \tan(\sigma_1\theta_1) & 1 \end{bmatrix}\cdots\begin{bmatrix} 1 & -\tan(\sigma_N\theta_N) \\ \tan(\sigma_N\theta_N) & 1 \end{bmatrix}$$

$$K = \cos(\sigma_0\theta_0)\cos(\sigma_1\theta_1)\cdots\cos(\sigma_N\theta_N) \qquad \text{scale factor}$$

$$\theta = \sigma_0 \theta_0 + \sigma_1 \theta_1 + \cdots + \sigma_N \theta_N \qquad \sigma_k \in \{-1, 0, +1\}$$

$$\theta'' = \boxed{\sum_{k=1}^{N} b_k \cdot 2^{-k} = \phi_0 + \sum_{k=2}^{N+1} r_k \cdot 2^{-k}} \qquad \begin{array}{l} b_k \in \{0, 1\} \\ r_k \in \{-1, +1\} \end{array}$$

$$K \begin{bmatrix} 1 & -\tan(\sigma_0\theta_0) \\ \tan(\sigma_0\theta_0) & 1 \end{bmatrix} \begin{bmatrix} 1 & -\tan(\sigma_1\theta_1) \\ \tan(\sigma_1\theta_1) & 1 \end{bmatrix} \cdots \begin{bmatrix} 1 & -\tan(\sigma_N\theta_N) \\ \tan(\sigma_N\theta_N) & 1 \end{bmatrix}$$

$$K = \cos(\theta_0)\cos(\theta_1)\cdots\cos(\theta_N) \qquad \theta_k = \tan^{-1} 2^{-k}$$

$$K \begin{bmatrix} 1 & -\tan(r_2\theta_2) \\ \tan(r_2\theta_2) & 1 \end{bmatrix} \begin{bmatrix} 1 & -\tan(r_3\theta_3) \\ \tan(r_3\theta_3) & 1 \end{bmatrix} \cdots \begin{bmatrix} 1 & -\tan(r_N\theta_N) \\ \tan(r_N\theta_N) & 1 \end{bmatrix}$$

$$K = \cos(\theta_2)\cos(\theta_3)\cdots\cos(\theta_N) \qquad \theta_k = 2^{-k}$$

The recoding maintains a constant scale factor K

$$K \begin{bmatrix} 1 & -\tan(r_2\theta_2) \\ \tan(r_2\theta_2) & 1 \end{bmatrix} \begin{bmatrix} 1 & -\tan(r_3\theta_3) \\ \tan(r_3\theta_3) & 1 \end{bmatrix} \cdots \begin{bmatrix} 1 & -\tan(r_N\theta_N) \\ \tan(r_N\theta_N) & 1 \end{bmatrix}$$

$$K = \cos(\theta_2)\cos(\theta_3)\cdots\cos(\theta_N) \qquad \theta_k = 2^{-k}$$

$$\begin{bmatrix} X_{k+1} \\ Y_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & -\tan(r_k\theta_k) \\ \tan(r_k\theta_k) & 1 \end{bmatrix} \begin{bmatrix} X_k \\ Y_k \end{bmatrix}.$$

$$= \begin{bmatrix} X_k - \tan(r_k\theta_k)Y_k \\ Y_k + \tan(r_k\theta_k)X_k \end{bmatrix}$$

Sub rotation

$$X_{k+1} = X_k - \tan(r_k\theta_k)Y_k$$

$$Y_{k+1} = Y_k + \tan(r_k\theta_k)X_k$$