Hybrid CORDIC 4.A

20170527

radix-2 decomposition

any rotation angle θ

~ linear combination of angles

◎ the radix-2 based set

$\{2^{-i}\}, \quad i = 1, 2, \cdots, n-1$

$\sum_{i=0}^{n-1} b_i 2^{-i}, \quad b_i \in \{0, 1\}$

↳ determines whether a micro-rotation is to be performed or not.

- not widely used ($\Leftarrow$ no hardware benefit)

◎ the elementary angle set

$\{\tan^{-1} 2^{-i}\}, \quad i = 1, 2, \cdots, n-1$

- can be used in implementing shift-and-add operations

$\{2^{-i} : i \in \{1, 2, \cdots, n-1\}\}$

$\{\tan^{-1}(2^{-i}) : i \in \{1, 2, \cdots, n-1\}\}$

Coarse - Fine Decomposition

for sufficiently large j

$\alpha_j = \tan^{-1}(2^{-j}) \approx 2^{-j}$ when $\boxed{j > \lceil \frac{n}{3} \rceil - 1}$

the elementary angle set

$\{\tan^{-1}(2^{-j})\}$ for most significant part
$\{(2^{-j})\}$ for less significant part

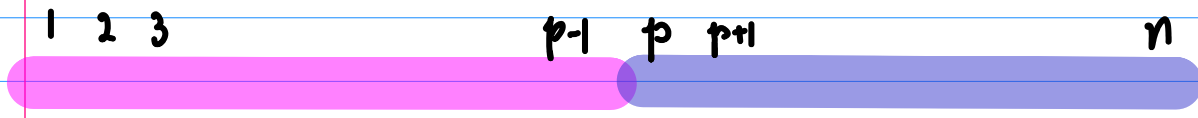
radix set $S = S_1 \cup S_2$

$S_1 = \{\tan^{-1}(2^{-i}) : i \in [1, 2, \cdots, p-1]\}$

$S_2 = \{(2^{-i}) : i \in [p, p+1, \cdots, n-1]\}$

the rotation angle is partitioned

$$\theta = \theta_H + \theta_L$$



$S_1 = \{ \tan^{-1}(2^{-i}) \}$ $\qquad$ $S_2 = \{ 2^{-i} \}$

$p > \lceil \frac{n}{3} \rceil - 1$

$$\theta = \underset{S_1}{\underset{\cap\!\!\!\!\!\mid}{\theta_H}} + \underset{S_2}{\underset{\cap\!\!\!\!\!\mid}{\theta_L}}$$

Wang & Swartzlander, Hybrid CORDIC algorithms
IEEE Computers, 1997

rotation angle decomposition

$$\theta = \theta_H + \theta_L$$ coarse & fine subangles

$$\theta_H = \sum_{i=1}^{p-1} \sigma_i \tan^{-1} 2^{-i}$$ $\sigma_i \in \{1, -1\}$

$$\theta_L = \sum_{i=p}^{n-1} d_i \, 2^{-i}$$ $d_i \in \{0, 1\}$

Wang & Swartzlander, Hybrid CORDIC algorithms
IEEE Computers, 1997

ATR (Arc Tangent Radix)

$$\{\alpha_0, \alpha_1, \cdots, \alpha_{N-1}\} = \{\tan^{-1} 2^0, \tan^{-1} 2^{-1}, \cdots, \tan^{-1} 2^{-N+1}\}$$

CORDIC convergence theorem

$$\alpha_i - \sum_{j=i+1}^{N-1} \alpha_j < \alpha_{N-1}$$

The Hybrid Radix sets

Mixed-Hybrid Circular ATR

$$\{\underbrace{\tan^{-1} 2^0, \tan^{-1} 2^{-1}, \cdots, \tan^{-1} 2^{-n+1}}_{\text{most significant part, } n\text{-bits}}, \underbrace{2^{-n}, 2^{-n+1}, \cdots, 2^{-N+1}}_{\text{least significant, } N-n \text{ bits}}\}$$

most significant part

n-bits

least significant

N − n bits

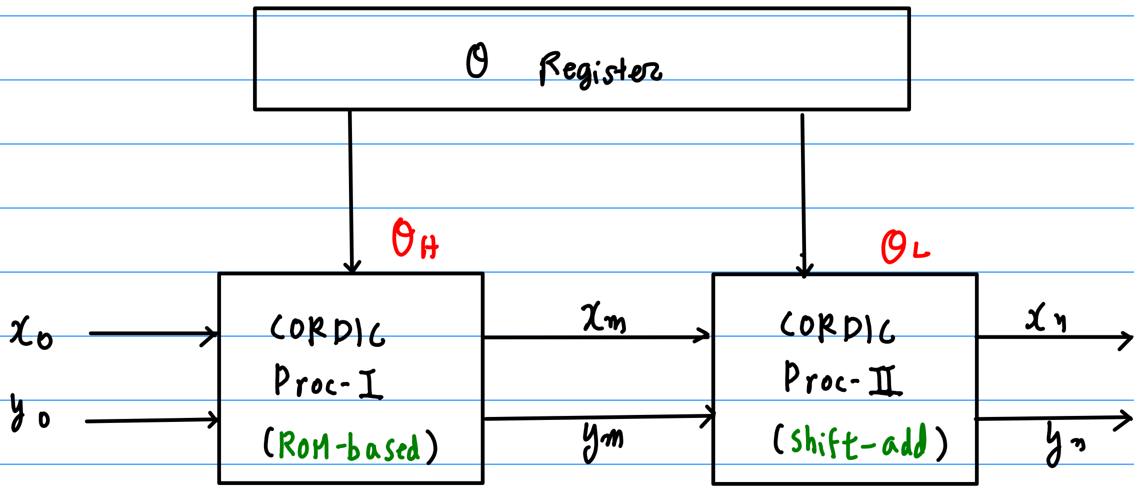
$$\theta_H = \sum_{i=0}^{n-1} \theta_i 2^{-i} \qquad \theta_L = \sum_{i=n}^{N-1} \theta_i 2^{-i}$$

Partitioned-Hybrid Circular ATR

$$\{\underbrace{\tan^{-1} 2^{-n+1}}_{\text{most significant part}}, \underbrace{2^{-n}, 2^{-n+1}, \cdots, 2^{-N+1}}_{\text{least significant}}\}$$

most significant part

least significant



two cascade stages

Coarse rotation

$$\begin{bmatrix} x_m \\ y_m \end{bmatrix} = \begin{bmatrix} 1 & -\tan(\theta_H) \\ \tan(\theta_H) & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$$

Fine rotation

$$\begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & -\tan(\theta_L) \\ \tan(\theta_L) & 1 \end{bmatrix} \begin{bmatrix} x_m \\ y_m \end{bmatrix}$$

$$\begin{bmatrix} x_n \\ y_n \end{bmatrix} \longleftarrow \begin{bmatrix} x_m \\ y_m \end{bmatrix} \longleftarrow \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$$

Implementation of Hybrid CORDIC

CORDIC PROC-I : the coarse rotation ROM based

θ_H { ROM look-up operation
addition

CORDIC PROC-II : the fine rotation SHIFT-ADD

θ_L sequence of shift-and-add
no computation of the direction of micro-rotation
the need of a micro-rotation is explicit $(=d_i)$
in the radix-2 representation

$$\theta_L = \sum_{i=p}^{n-1} d_i \, 2^{-i} \qquad d_i \in \{0, 1\}$$

* in the signed digit notation [Timmermann Low Latency]

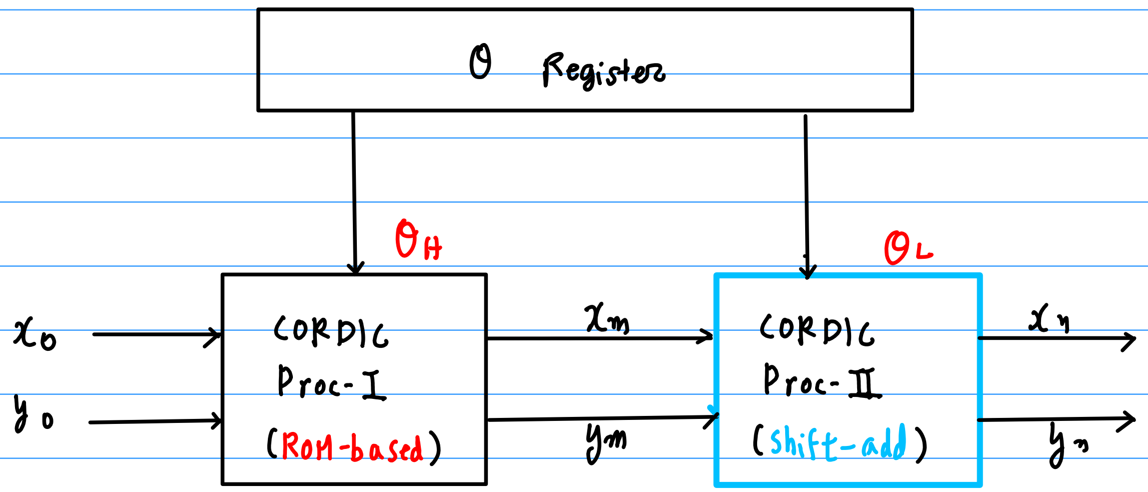
$$\theta_L = \sum_{i=p}^{n-1} \tilde{b}_i \, 2^{-i} \qquad \tilde{b}_i = \{-1, +1\}$$

$$\theta_L = \sum_{i=p}^{n-1} d_i \, 2^{-i} \qquad d_i \in \{0, 1\}$$

$$= \sum_{i=p}^{n-1} \tilde{b}_i \, 2^{-i} \qquad \tilde{b}_i = \{-1, +1\}$$

① radix-2 representation

② signed digit notation



* the direction is explicit
  → parallel implementation possible

Timmermann, Low Latency time CORDIC algorithm, 1992

* the hybrid decomposition could be used

  ROM-based realization of coarse rotation
  → minimize latency

  Shift-and-add implementation of fine rotation
  → minimize hardware complexity
  ← no need to find the rotation direction

[23] M. Kuhlmann and K. K. Parhi, "P-CORDIC: A precomputation based rotation CORDIC algorithm," *EURASIP J. Appl. Signal Process.*, vol. 2002, no. 9, pp. 936–943, 2002.

very high precision

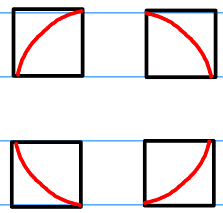
ROM size $n \cdot 2^{n/5}$ bits

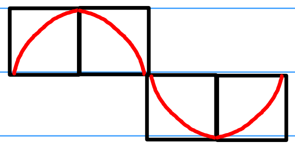
if latency is tolerable
the conventional CORDIC
shift-and-add operation

Shift-Add Implementation of Coarse rotation

Using the symmetry properties
of cosine and sine functions
in different quadrants

rotation through arbitrary angle θ
can be mapped from $[0, 2\pi]$
to the first half of the first quadrant $[0, \frac{\pi}{4}]$





the coarse-fine partition
could be applied for reducing
the number of micro-rotations
necessary for fine rotations

to implement the coarse rotation
through shift-add operation

the coarse sub angle θ_M
in terms of elementary rotations of the form $\tan^{-1} 2^{-i}$

$$\theta_M = \sum_{i=1}^{p-1} d_i \, 2^{-i} \qquad = \sum_{i=1}^{p-1} \left(\sigma_i \tan^{-1}(2^{-i}) - \theta_{L_i} \right)$$

$$\sum_{i=1}^{p-1} \boxed{d_i \, 2^{-i}} \qquad \qquad \sum_{i=1}^{p-1} \boxed{\sigma_i \tan^{-1}(2^{-i})}$$

shift-add $\qquad$ coarse rotation

θ_{L_i} : correction term

$$\theta = \theta_H + \theta_L$$

$$= \theta_M + \tilde{\theta}_L$$

$$= \sum_{i=1}^{p-1} \sigma_i \tan^{-1}(2^{-i}) + \sum_{i=p}^{n-1} d_i 2^{-i}$$

$$= \sum_{i=1}^{p-1} d_i 2^{-i} + \boxed{\sum_{i=1}^{p-1} \theta_{L_i} + \sum_{i=p}^{n-1} d_i 2^{-i}}$$

$$\boxed{\sum_{i=1}^{p-1} \theta_{L_i} + \theta_L}$$

$$\boxed{\tilde{\theta}_L}$$

$$\theta = \boxed{\theta_H} + \boxed{\theta_L} \quad \text{(higher \& lower parts)}$$

$$= \boxed{\sum_{i=1}^{p-1} \sigma_i \tan^{-1} 2^{-i}} + \boxed{\sum_{i=p}^{n-1} d_i \, 2^{-i}}$$

$$\sigma_i \in \{1, -1\} \qquad d_i \in \{0, 1\}$$

$$\boxed{\theta_M} + \boxed{\tilde{\theta}_L}$$

$$= \boxed{\sum_{i=1}^{p-1} \sigma_i \tan^{-1} 2^{-i} - \sum_{i=1}^{p-1} \theta_{Li}} + \boxed{\sum_{i=1}^{p-1} \theta_{Li} + \sum_{i=p}^{n-1} d_i \, 2^{-i}}$$

$$= \boxed{\sum_{i=1}^{p-1} d_i \, 2^{-i}} + \boxed{\sum_{i=1}^{p-1} \theta_{Li} + \sum_{i=p}^{n-1} d_i \, 2^{-i}}$$

$$\tilde{\theta}_L = \theta_L + \sum_{i=2}^{N/3} \theta_{L_i}$$

$$\boxed{(-\theta_0) + \sum_{j=1}^{m-1} \theta_j 2^{j-1}} + \boxed{\sum_{j=m}^{N} \theta_j 2^{-j}}$$

$$\theta_M \simeq \sum_{i=2}^{N/3} d_i\, 2^{-i} = \sum_{i=2}^{N/3} \left(\sigma_i \tan^{-1}(2^{-i}) + \theta_{L_i}\right)$$

[24] D. Fu and A. N. Willson, Jr., "A high-speed processor for digital sine/cosine generation and angle rotation," in *Conf. Rec. 32nd Asilomar Conf. on Signals, Syst. & Computers*, Nov. 1998, vol. 1, pp. 177–181.

[25]

[25] C.-Y. Chen and W.-C. Liu, "Architecture for CORDIC algorithm realization without ROM lookup tables," in *Proc. 2003 Int. Symp. on Circuits Syst., ISCAS'03*, May 2003, vol. 4, pp. 544–547.

both coarse and fine rotations
can be implemented by a sequence of
shift-add operations
without ROM look-up or
real multiplication

PROC-I : the conventional CORDIC
the first 1/3 iteration

{ the residual angle
the intermediate rotated vector

reduced latency implementation
sine-cosine generation

[24] D. Fu and A. N. Willson, Jr., “A high-speed processor for digital sine/cosine generation and angle rotation,” in *Conf. Rec. 32nd Asilomar Conf. on Signals, Syst. & Computers*, Nov. 1998, vol. 1, pp. 177–181.

[25] C.-Y. Chen and W.-C. Liu, “Architecture for CORDIC algorithm realization without ROM lookup tables,” in *Proc. 2003 Int. Symp. on Circuits Syst., ISCAS’03*, May 2003, vol. 4, pp. 544–547.

[26] D. Fu and A. N. Willson, Jr., “A two-stage angle-rotation architecture and its error analysis for efficient digital mixer implementation,” *IEEE Trans.Circuits Syst. I: Reg. Papers*, vol. 53, no. 3, pp. 604–614, Mar. 2006.

[27] S. Ravichandran and V. Asari, “Implementation of unidirectional CORDIC algorithm using precomputed rotation bits,” in *45th Midwest Symp. on Circuits Syst., 2002. MWSCAS 2002*, Aug. 2002, vol. 3, pp. 453–456.

[28] C.-Y. Chen and C.-Y. Lin, “High-resolution architecture for CORDIC algorithm realization,” in *Proc. Int. Conf. on Commun., Circuits Syst., ICCCS’06*, June 2006, vol. 1, pp. 579–582.

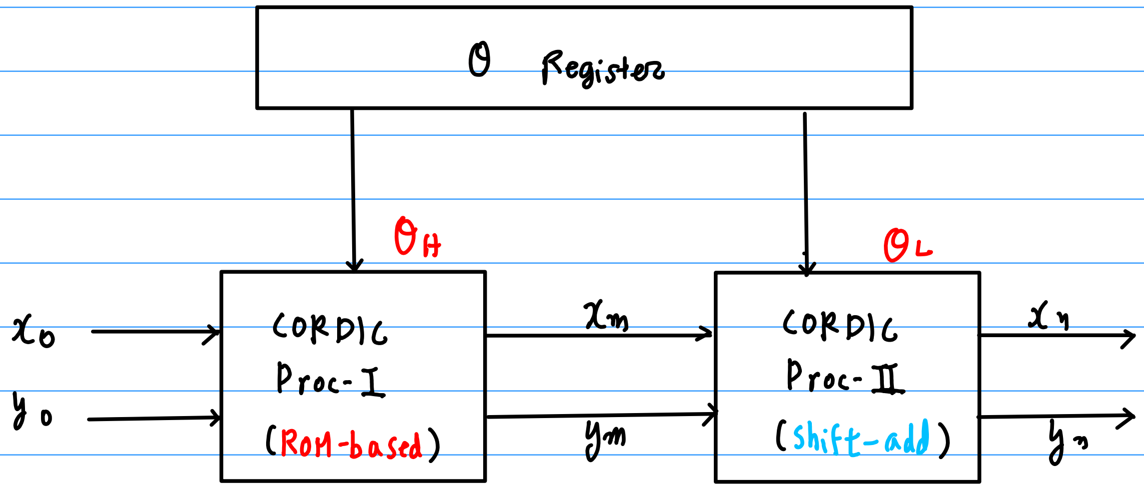
High speed, High precision [24], [26]

[29] D. D. Hwang, D. Fu, and A. N. Willson, Jr., “A 400-MHz processor for the conversion of rectangular to polar coordinates in 0.25-/mu-m CMOS,” *IEEE J. Solid-State Circuits*, vol. 38, no. 10, pp. 1771–1775, Oct. 2003.

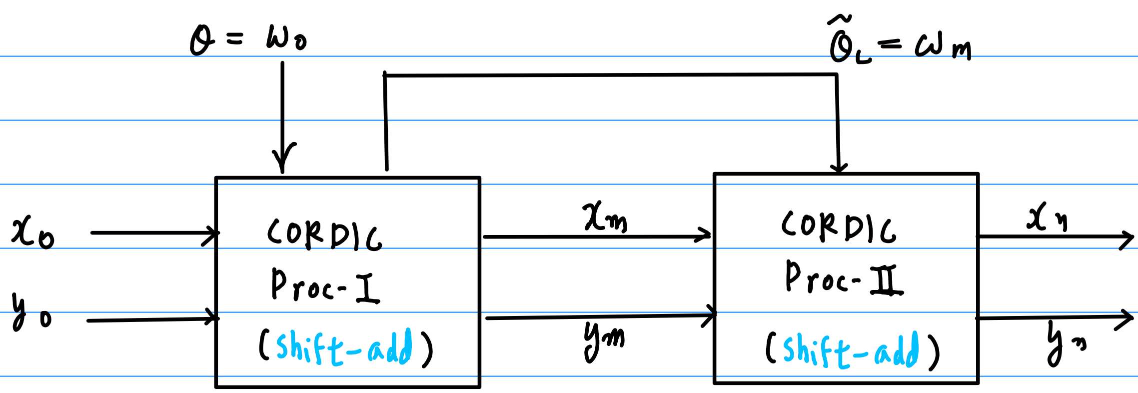
[30] S.-W. Lee, K.-S. Kwon, and I.-C. Park, “Pipelined cartesian-to-polar coordinate conversion based on SRT division,” *IEEE Trans. Circuits Syst. II: Express Briefs*, vol. 54, no. 8, pp. 680–684, Aug. 2007.

ordinates

[10] Hybrid CORDIC, Wang, Swartzlander



[24] Fu & Willson Sine / Cosine Generation



$$\theta_H = \sum_{i=1}^{p-1} d_i 2^{-i} = \sum_{i=1}^{p-1} \left(\sigma_i \tan^{-1} 2^{-i} - \underbrace{\theta_{L_i}}_{\text{correction term}}\right)$$

$$\theta = \sum_{i=1}^{p-1} d_i 2^{-i} + \tilde{\theta}_L$$

$$\tilde{\theta}_L = \theta_L + \sum_{i=1}^{p-1} \theta_{L_i}$$

Wilson ROM based Sine/Cosine Generation

[24] Fu & Willson Sine / Cosine Generation

ROM-based

For high resolution, ROM size grows exponentially

quater-wave symmetry

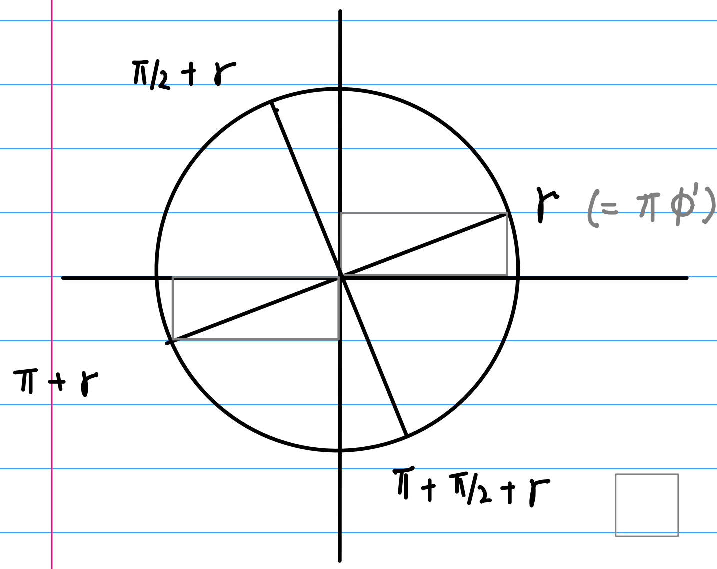
$$\sin\theta = \cos\left(\frac{\pi}{2} - \theta\right)$$

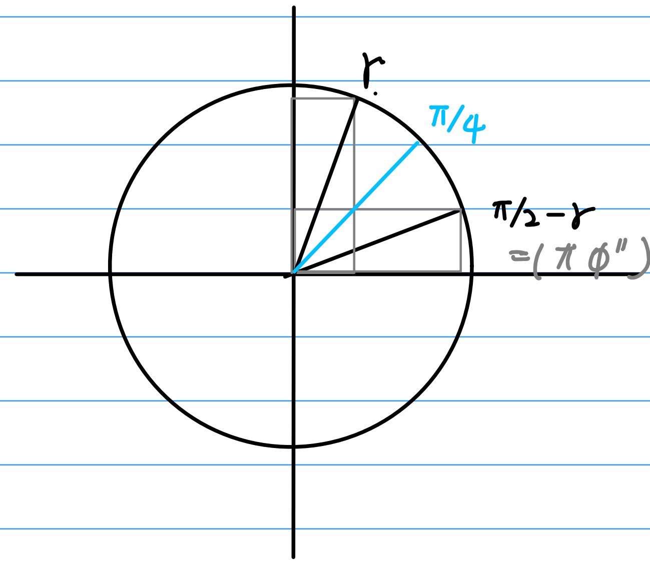
ϕ $[0, 2\pi] \longrightarrow [0, \frac{\pi}{4}]$

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

$$\underset{\phi}{[-1, 1]} \longrightarrow \underset{\theta}{[0, \pi/4]} \longrightarrow \text{sine/cosine generator}$$

$$\pi\phi \longleftarrow$$

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

2 msb of the normalized angle ϕ

: MSB_1, MSB_2 determine the quadrant of $\pi\phi$
MSB_3 determines the upper / lower half of the quadrant

Controe / Interchange

$MSB_1 \leftarrow 0$, $MSB_2 \leftarrow 0$ $\pi\phi'$

$MSB_3 = 1$: the upper half quadrant
$\sin r = \cos(\pi/2 - r)$
$\cos r = \sin(\pi/2 - r)$
the normalized angle below $\pi/4$ ϕ''
$\phi'' = 0.5 - \phi'$ $(MSB_3 = 1)$
$\phi'' = \phi'$ $(MSB_3 = 0)$

$\theta = \pi\phi''$ hardwired multiplier
$[0, \pi/4]$

⌈ CORDIC-based architectures
the elementary angles are divided by π

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

since the directions of the subrotations are controlled only by the sign of the difference between two angles and not the magnitude, the multiplication by π is not required ⌋

Output stage $\sin\theta \rightarrow \sin\pi\phi$ $\quad [-\pi, +\pi]$

$\cos\theta \rightarrow \cos\pi\phi$

negation / interchange

MSB_1
MSB_2
MSB_3

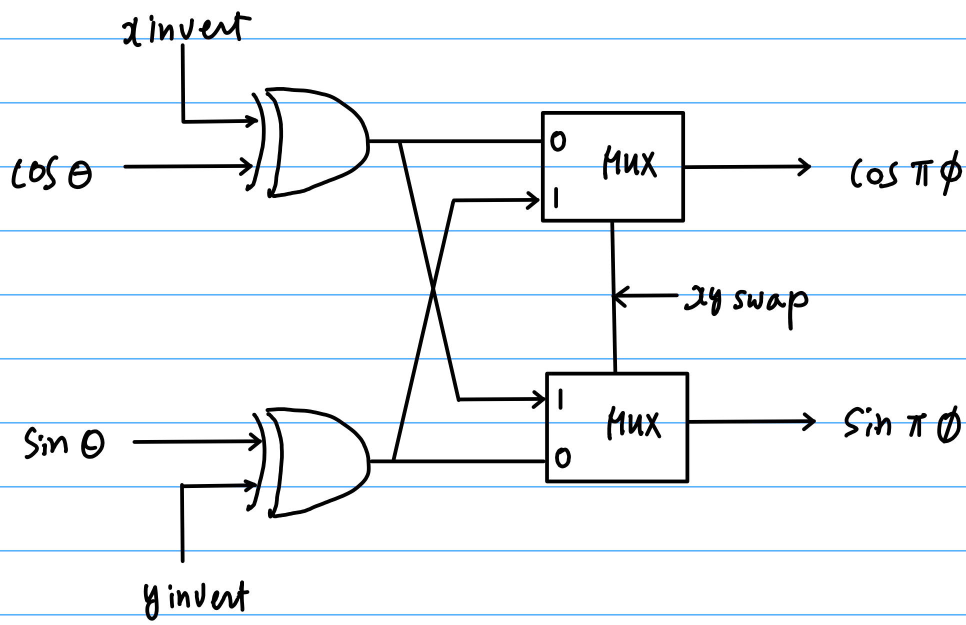
→ x invert
→ y invert
→ xy swap

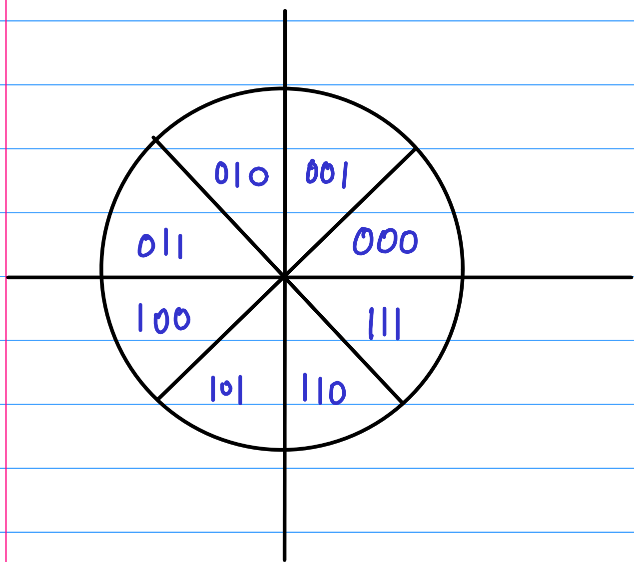
the negation of $\cos\theta = X_{N+1}$

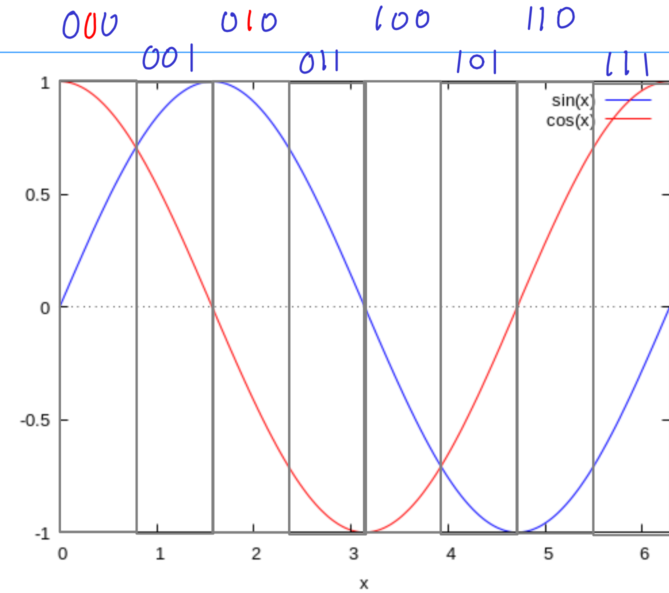
$\sin\theta = Y_{N+1}$

Interchange

negate before swap







	Xinv	Yinv	swap	$\cos\pi\theta$	$\sin\pi\phi$
000	0	0	0	$\cos\theta$	$\sin\theta$
001	0	0	1	$\sin\theta$	$\cos\theta$
010	0	1	1	$-\sin\theta$	$\cos\theta$
011	1	0	0	$-\cos\theta$	$\sin\theta$
100	1	1	0	$-\cos\theta$	$-\sin\theta$
101	1	1	1	$-\sin\theta$	$-\cos\theta$
110	1	0	1	$\sin\theta$	$-\cos\theta$
111	0	1	0	$\cos\theta$	$-\sin\theta$

Parallel CORDIC based on coarse and fine decomposition

[31] S.-F. Hsiao, Y.-H. Hu, and T.-B. Juang, "A memory-efficient and high-speed sine/cosine generator based on parallel CORDIC rotations," *IEEE Signal Process. Lett.*, vol. 11, no. 2, 2004.

Wisconsin Madison

2 angle recording techniques

parallel detection of the direction of micro rotations

* for the **coarse** part of the angle θ_H

BBR (Binary to Bipolar Recoding)

MAR (Micro-rotation Angle Recoding)

BBR used to obtain the polarity of each bit in the radix-2 representation of θ_H to determine the rotation direction.

MAR used to decompose each positional binary weight 2^{-i}, $\forall i$ $i = 1, 2, \cdots, m-1$ into a linear combination of $\boxed{\tan^{-1}}$ terms

[32] T.-B. Juang, S.-F. Hsiao, and M.-Y. Tsai, "Para-CORDIC: Parallel CORDIC rotation algorithm," *IEEE Trans. Circuits Syst. I: Regular Papers*, vol. 51, no. 8, 2004.
[33] T.-B. Juang, "Area/delay efficient recoding methods for parallel CORDIC rotations," in *IEEE Asia Pacific Conf. on Circuits Syst., APCCAS'06*, Dec. 2006, pp. 1539–1542.

Binary to Bipolar Recording (BBR)

the initial angle

$$\theta = (-\theta_0) + \sum_{j=1}^{N} \theta_j 2^{-j} \qquad \theta_j \in \{0, 1\}$$

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

$$\tan^{-1} 2^{-i} = 2^{-i} \qquad i \geqslant m = \lceil (N - \log_2 3)/3 \rceil$$

the last $2N/3$ rotation directions $\sigma_m \cdots \sigma_N$ can be obtained in parallel after the completion of the first $N/3$ iterations

$$\theta = \boxed{\theta_H} + \boxed{\theta_L} \quad \text{(higher \& lower parts)}$$

$$= \boxed{(-\theta_0) + \sum_{j=1}^{m-1} \theta_j 2^{j-1}} + \boxed{\sum_{j=m}^{N} \theta_j 2^{-j}}$$

$$\theta_H = (-\theta_0) + \sum_{j=1}^{m-1} \left[2^{-j-1} + (2\theta_j - 1) 2^{-j-1} \right]$$

$j=1$ $\quad 2^{-2} + (2\theta_1 - 1)\, 2^{-2} \qquad 2^{-1-1} = 2^{-2}$

$j=2$ $\quad 2^{-3} + (2\theta_2 - 1)\, 2^{-3} \qquad 2^{-2-1} = 2^{-3}$

$j=3$ $\quad 2^{-4} + (2\theta_3 - 1)\, 2^{-4} \qquad 2^{-3-1} = 2^{-4}$

$\vdots$

$j=m-1$ $\quad 2^{-m} + (2\theta_{m-1} - 1)\, 2^{-m} \qquad 2^{-m+1-1} = 2^{-m}$

$$\sum_{k=2}^{m} r_k 2^{-k}$$

$$\frac{a(1-r^n)}{1-r}$$

$$r_k = (2\theta_{k-1} - 1)$$

$$= \frac{2^{-2}(1-(2^{-1})^{m-1})}{1-2^{-1}}$$

$\theta_{k-1} = 0 \qquad r_k \to -1$

$\theta_{k-1} = 1 \qquad r_k \to +1$

$$= \frac{2^{-2}(1-2^{-m+1})}{2^{-1}}$$

$$= 2^{-1}(1-2^{-m+1})$$

$$= 2^{-1} - 2^{-m}$$

$\theta = \boxed{\theta_H} + \boxed{\theta_L}$ (higher & lower parts)

$$= \boxed{(-\theta_0) + \sum_{j=1}^{m-1} \theta_j 2^{j-1}} + \boxed{\sum_{j=m}^{N} \theta_j 2^{-j}}$$

$$\boxed{\begin{array}{l} \theta_H = (-\theta_0) + 2^{-1} + \sum_{k=2}^{m} r_k 2^{-k} - 2^{-m} \\ r_k = (2\theta_k - 1) \qquad r_k \in \{-1, +1\} \end{array}}$$

BBR for θ_j

$j = 0, 1, 2, \cdots, m-1$

* Binary to Bipolar Recoding

initial input angle

$$\theta = (-\theta_0) + \sum_{j=1}^{N} \theta_j 2^{-j}$$

binary

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

higher part angle

$$\theta_H = (-\theta_0) + 2^{-1} + \sum_{k=2}^{m} r_k 2^{-k} \quad 2^{-m}$$

bipolar

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

$r_k = (2\theta_k - 1)$

Microrotation Angle Recording

$i = 1, \cdots, m-1 \Rightarrow \tan^{-1}(2^{-i}) \neq 2^{-i}$

decompose each positional binary weight

$\boxed{2^{-i}}$, $i = 1, 2, \cdots, m-1$ into

$\begin{cases} \text{the combination of significant } \boxed{\tan^{-1}(2^{-j})} \text{ terms} \\ \text{plus error terms } \boxed{e_i} \end{cases}$

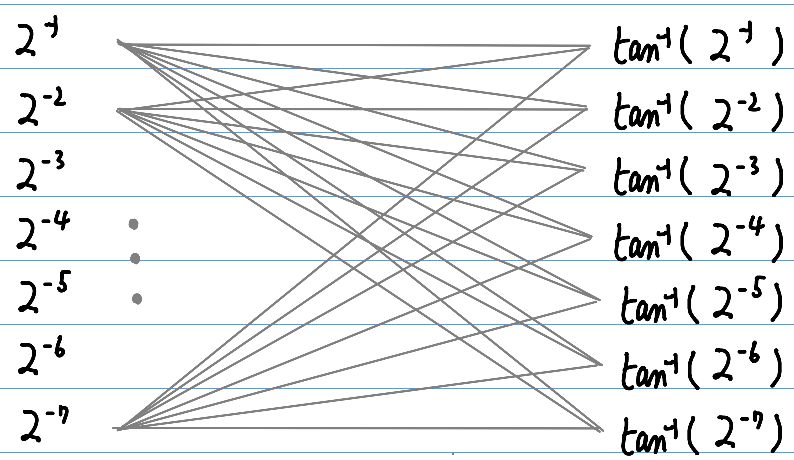
$\begin{cases} \text{Collecting all the other} \\ \text{insignificant values of} \\ \tan^{-1}(2^{-j}),\ j > m \end{cases}$

$N = 24, \Rightarrow m = \lceil (N - \log_2 3)/3 \rceil = 8$

$m = 8$

micro rotation angle

2^{-i} as a linear combination of $\tan^{-1}(2^{-j})$



$2^{-1} = \tan^{-1}(2^{-1}) + \tan^{-1}(2^{-5}) + \tan^{-1}(2^{-8}) + e_1$

$2^{-2} = \tan^{-1}(2^{-2}) + \tan^{-1}(2^{-8}) + e_2$

$2^{-3} = \tan^{-1}(2^{-3}) + e_3$

$2^{-4} = \tan^{-1}(2^{-4}) + e_4$

$2^{-5} = \tan^{-1}(2^{-5}) + e_5$

$2^{-6} = \tan^{-1}(2^{-6}) + e_6$

$2^{-7} = \tan^{-1}(2^{-7}) + e_7$

$$\theta_H = (-\theta_0) + 2^{-1} + \sum_{k=2}^{m} r_k 2^{-k} - 2^{-m}$$

$$= (-\theta_0) + 2^{-1} + \sum_{k=2}^{8} r_k 2^{-k} - 2^{-8}$$

$$= (1-2\theta_0) 2^{-1} + \sum_{k=2}^{8} r_k 2^{-k} - 2^{-8}$$

$$r_k = (2\theta_k - 1) \qquad r_k \in \{-1, +1\}$$

the first 8 rotation directions are computed concurrently

$$\sigma_1 = (1-2\theta_0)$$

$$\sigma_k = r_k = (2\theta_{k-1} - 1), \qquad k = 2, \cdots, 8$$

$$\sigma_1 = (2\theta_0 - 1)\cdot(-1)$$

$$\sigma_2 = (2\theta_1 - 1)$$

$$\sigma_3 = (2\theta_2 - 1)$$

$$\sigma_4 = (2\theta_3 - 1)$$

$$\sigma_5 = (2\theta_4 - 1)$$

$$\sigma_6 = (2\theta_5 - 1)$$

$$\sigma_7 = (2\theta_6 - 1)$$

$$\sigma_8 = (2\theta_7 - 1)$$

$$\theta_{17} = (-\theta_0) + 2^{-1} + \sum_{k=2}^{m} r_k 2^{-k} - 2^{-m}$$

$$= (1-2\theta_0)2^{-1} + \sum_{k=2}^{m} r_k 2^{-k} - 2^{-m}$$

$$= (1-2\theta_0)2^{-1} + \sum_{k=2}^{8} r_k 2^{-k} - 2^{-8}$$

$$r_k = (2\theta_{k-1} - 1) \in \{-1, +1\}$$

the first 8 rotation directions

$$\sigma_1 = (1-2\theta_0)$$
$$\sigma_2 = (2\theta_1 - 1) = r_2$$
$$\sigma_3 = (2\theta_2 - 1) = r_3$$
$$\sigma_4 = (2\theta_3 - 1) = r_4$$
$$\sigma_5 = (2\theta_4 - 1) = r_5$$
$$\sigma_6 = (2\theta_5 - 1) = r_6$$
$$\sigma_7 = (2\theta_6 - 1) = r_7$$
$$\sigma_8 = (2\theta_7 - 1) = r_8$$

all the signed error terms $\sigma_i e_i$ $i = 1, \cdots, 7$
and the last term -2^{-8} are added to θ_L

the corrected lower part $\hat{\theta}_L$

2's complement form

$$\hat{\theta}_L = \theta_L + \sum_{i=1}^{7} \sigma_i e_i - 2^{-8}$$

$$= (-\hat{\theta}_7)\, 2^{-7} + \sum_{k=8}^{24} \hat{\theta}_k 2^{-k}$$

$\hat{\theta}_k \in \{0, 1\}$

$k = 8, \cdots, 24$

$$2^{-1} = \tan^{-1}(2^{-1}) + \tan^{-1}(2^{-5}) + \tan^{-1}(2^{-8}) \quad + e_1 \times \sigma_1$$
$$2^{-2} = \tan^{-1}(2^{-2}) + \tan^{-1}(2^{-8}) \quad + e_2 \times \sigma_2$$
$$2^{-3} = \tan^{-1}(2^{-3}) \quad + e_3 \times \sigma_3$$
$$2^{-4} = \tan^{-1}(2^{-4}) \quad + e_4 \times \sigma_4$$
$$2^{-5} = \tan^{-1}(2^{-5}) \quad + e_5 \times \sigma_5$$
$$2^{-6} = \tan^{-1}(2^{-6}) \quad + e_6 \times \sigma_6$$
$$2^{-7} = \tan^{-1}(2^{-7}) \quad + e_7 \times \sigma_7$$

$$\theta_H = (1 - 2\theta_0)2^{-1} + \sum_{k=2}^{8} r_k 2^{-k} - 2^{-8}$$

$+$

$\rightarrow \theta_L$

Corrected lower part angle

$$\hat{\theta}_L = \theta_L + \sum_{i=1}^{7} e_i \cdot \sigma_i - 2^{-8}$$

all the signed error terms $\sigma_i e_i$ $i = 1, \cdots, 7$
and the last term -2^{-8}
are added to θ_L
generating the corrected lower part $\hat{\theta}_L$

$$\hat{\theta}_L = \theta_L + \sum_{i=1}^{7} e_i \cdot \sigma_i - 2^{-8}$$

$$= (1 - \hat{\theta}_7) 2^{-7} + \sum_{k=8}^{24} \hat{\theta}_k 2^{-k}, \qquad \hat{\theta}_k \in \{0, 1\}$$

$$k = 8, \cdots, 24$$

$$|\hat{\theta}_L| \leq 2^{-7}$$

$$\tan^{-1} 2^{-i} = 2^{-i} \qquad i \geq 8$$

$$\hat{\theta}_L = \theta_L + \sum_{i=1}^{7} e_i \cdot \sigma_i - 2^{-8}$$

$$= (-\hat{\theta}_7) 2^{-7} + \sum_{k=8}^{24} \hat{\theta}_k 2^{-k}, \qquad \hat{\theta}_k \in \{0, 1\}$$

$$k = 8, \cdots, 24$$

$$\hat{\theta}_L = (-\hat{\theta}_7) 2^{-7} + \sum_{k=8}^{24} \hat{\theta}_k 2^{-k}$$

$$= (-\hat{\theta}_7) 2^{-7} + \sum_{k=9}^{25} (2\theta_{k-1} - 1) 2^{-k} + 2^{-8} - 2^{-25}$$

$$= (1 - 2\hat{\theta}_7) 2^{-8} + \sum_{k=9}^{25} \hat{r}_k 2^{-k} - 2^{-25}$$

$$\hat{r}_k = (2\hat{\theta}_k - 1) \in \{1, -1\}$$

$$\hat{\sigma}_8 = (1 - 2\hat{\theta}_7)$$

$$\hat{\sigma}_k = \hat{r}_k = (2\hat{\theta}_k - 1), \qquad k = 9, \cdots, 25$$

for the last $2N/3$ iterations

CSA Carry Save Adder

4:2 compressor