Lookahead CORDIC Literature

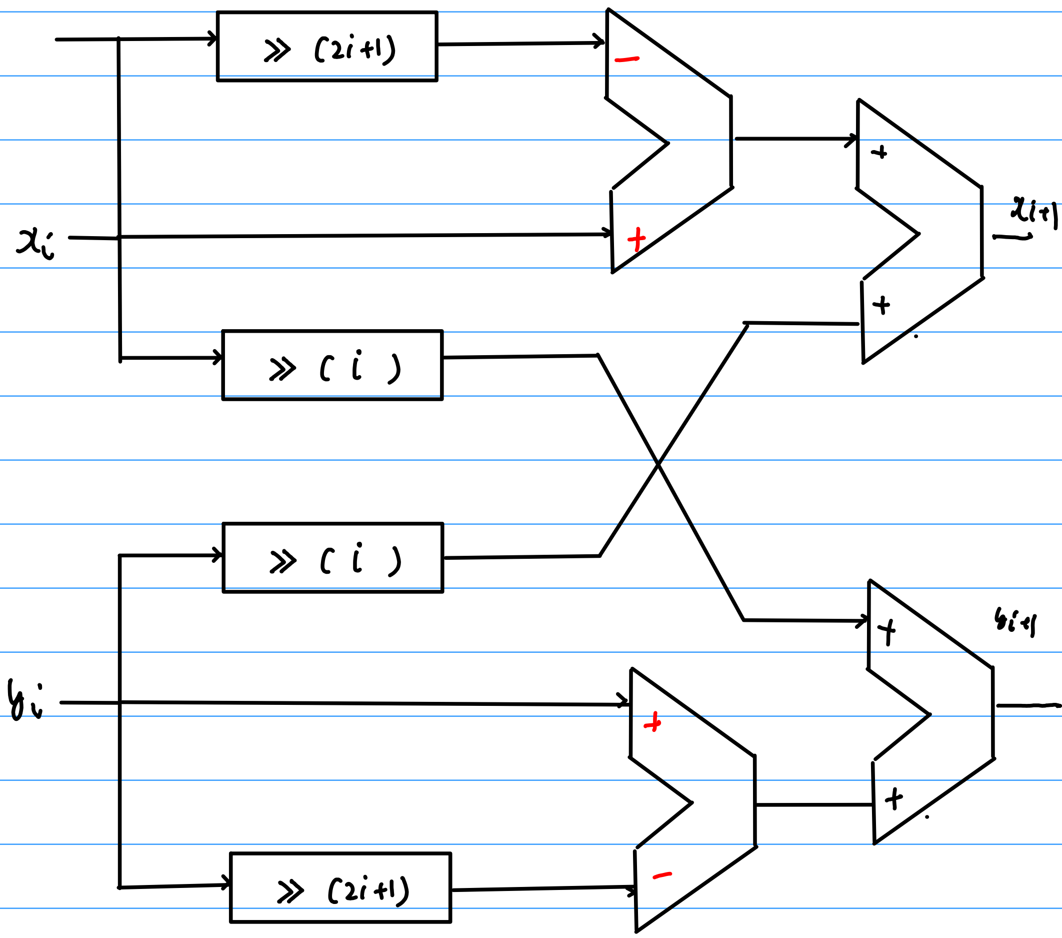
20160723

.

Lookahead Technique - CC Kao
Hybrid CORDIC - Wang & Swartzlander (1997)
Low Latency Time CORDIC Algorithms - Timmermann (1992)
Merged CORDIC Algorithms - Wang & Swartzlander (1995)
Merged Scaling Multiplication CORDIC Algorithms - Wang & Swartzlander (1997)
- Takagi (1987)
Redundant and on-line CORDIC - Ercegovac & Lang (1990)
Double Step Branching CORDIC - Phatak (1998)
- Duprat & Muller (1993)

Virtually scaling-free Maharatna

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \prod_{i=p}^{b-1} \begin{bmatrix} 1 - 2^{-(2i+1)} & 2^{-i} \\ -2^{-i} & -1 + 2^{-(2i+1)} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix}$$



$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 - 2^{-2i-1} & \mp(2^{-i} + 2^{-i-1}) \\ \pm(2^{-i} + 2^{-i-1}) & 1 - 2^{-2i-1} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \prod_{i=p}^{b-1} \begin{bmatrix} 1 - 2^{-(2i+1)} & 2^{-i} \\ -2^{-i} & -1 + 2^{-(2i+1)} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix}$$

Lookahead Technique

C.C. Kao

$$x_i = P_{x_i}\, x_0 - P_{y_i}\, y_0$$

$$y_i = P_{x_i}\, y_0 - P_{y_i}\, x_0$$

$$w_i = w_0 - \sigma_0 \alpha_0 - \cdots - \sigma_{i-1} \alpha_{i-1}$$

$$P_{x_1} = 2^{-0}\,(1)$$

$$P_{y_1} = 2^{-0}\,(1)$$

$$P_{x_2} = \left(1 - \frac{\sigma_0 \sigma_1}{2}\right)$$

$$P_{y_2} = \left(\sigma_0 + \frac{\sigma_1}{2}\right)$$

$$P_{x_3} = \left(1 - \frac{\sigma_0 \sigma_1}{2} - \frac{\sigma_0 \sigma_2}{4} - \frac{\sigma_1 \sigma_2}{8}\right)$$

$$P_{y_3} = \left(\sigma_0 + \frac{\sigma_1}{2} + \frac{\sigma_2}{4} - \frac{\sigma_0 \sigma_1 \sigma_2}{8}\right)$$

$$P_{x_4} = \left(1 - \frac{\sigma_0 \sigma_1}{2} - \frac{\sigma_0 \sigma_2}{4} - \frac{\sigma_1 \sigma_2}{8} - \frac{\sigma_0 \sigma_3}{8} - \frac{\sigma_1 \sigma_3}{16} - \frac{\sigma_2 \sigma_3}{32} + \frac{\sigma_0 \sigma_1 \sigma_2 \sigma_3}{64}\right)$$

$$P_{y_4} = \left(\sigma_0 + \frac{\sigma_1}{2} + \frac{\sigma_2}{4} - \frac{\sigma_0 \sigma_1 \sigma_2}{8} + \frac{\sigma_3}{8} - \frac{\sigma_0 \sigma_1 \sigma_3}{16} - \frac{\sigma_1 \sigma_2 \sigma_3}{32} - \frac{\sigma_1 \sigma_2 \sigma_3}{64}\right)$$

predict directly the rotation directions from the input angle

circuit parallelism

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} P_{x_1} & -P_{y_1} \\ P_{y_1} & P_{x_1} \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$$

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} P_{x_2} & -P_{y_2} \\ P_{y_2} & P_{x_2} \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$$

$$\begin{bmatrix} x_3 \\ y_3 \end{bmatrix} = \begin{bmatrix} P_{x_3} & -P_{y_3} \\ P_{y_3} & P_{x_3} \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$$

P_{x_i} P_{y_i} precoding matrices of x_i & y_i

stored in a <u>LUT</u>

① Wang, Piuri, Swartzlander (1997)

Hybrid CORDIC

the rotation directions

after the first m iterations

derived from the ω remainder — residual angles

at the end of the m-th iteration

with the conventional scheme

but execution time for ω operation

is reduced to $1/3$

N : significant word size (not including sign)

n : the first n iterations $\quad n = \left\lceil \dfrac{N - \log_2 3}{3} \right\rceil$

rotation directions can be computed

- in parallel
- without error

Hybrid Radix Sets

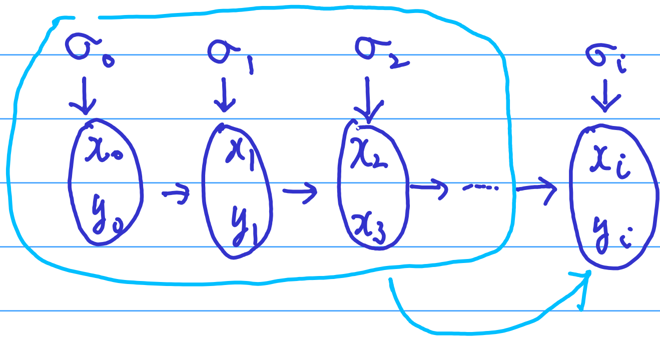
- ATR (Arc tangent Radix)
- radix 2

an arbitrary angle
- represented by a set of constant angles $\{\alpha_i\}$
  → acts like radix in a number system

the initial angle
θ - represented by a set of arc tangent constants

⇒ ATR : (Arc Tangent Radix)
$\{\alpha_0, \alpha_1, \alpha_2, \dots\} = \{\tan^{-1} 2^0, \tan^{-1} 2^{-1}, \tan^{-1} 2^{-2}, \dots\}$

$\sigma_0 \to \sigma_1 \to \sigma_2 \cdots \to \sigma_i$ sequential dependence



sequential dependence

parallelization bottleneck
1. rotation direction +, -
2. actual rotation of angles

* Timmerman 1992
Low Latency Time CORDIC Algorithms.

⇒ Merged CORDIC
- Swartzlander
- Shifter size reduced.

- merging 2 conventional CORDIC iterations in the same cycle

⇒ Timmerman 1992
Low Latency Time CORDIC Algorithms.

- parallel computation of rotation directions of a group

⇒ this paper

- partially parallelized

redundant adder 4-to-2

* Mixed Hybrid Circular ATR

$$\left\{ \{\text{most significant part}\}, \underline{\{\text{least significant part}\}} \right\}$$

→ parallelism exists

$$= \left\{ \tan^{-1} 2^{0}, \tan^{-1} 2^{-1}, \cdots, \tan^{-1} 2^{-n+1}, 2^{-n}, \cdots, 2^{-N+1} \right\}$$

* Partitioned - Hybrid Circular ATR

$$\left\{ \qquad \tan^{-1} 2^{-n+1}, 2^{-n}, \cdots, 2^{-N+1} \right\}$$

only one iteration

ROM-based LUT.

initial one at $n-1$ iteration

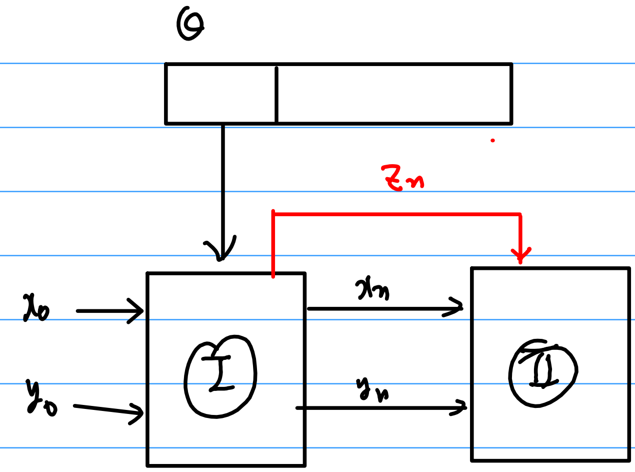
$$0\ 0\ \cdots\cdots\ 0\ 1\ x\ x\ \cdots\ x$$

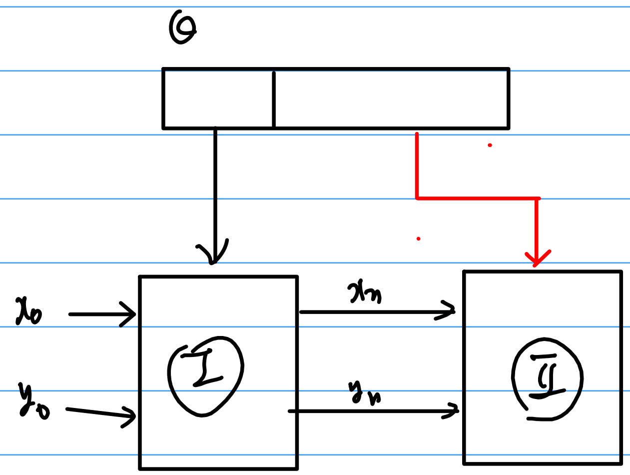
any possible angle θ_H

$$\sigma_H = \frac{\theta_H}{\tan^{-1} 2^{-n+1}} = \frac{\sum_{i=0}^{n-1} \theta_i 2^{-i}}{\tan^{-1} 2^{-n+1}}$$

Lookahead

* Mixed - Hybrid CORDIC





I
- ROM
- traditional CORDIC

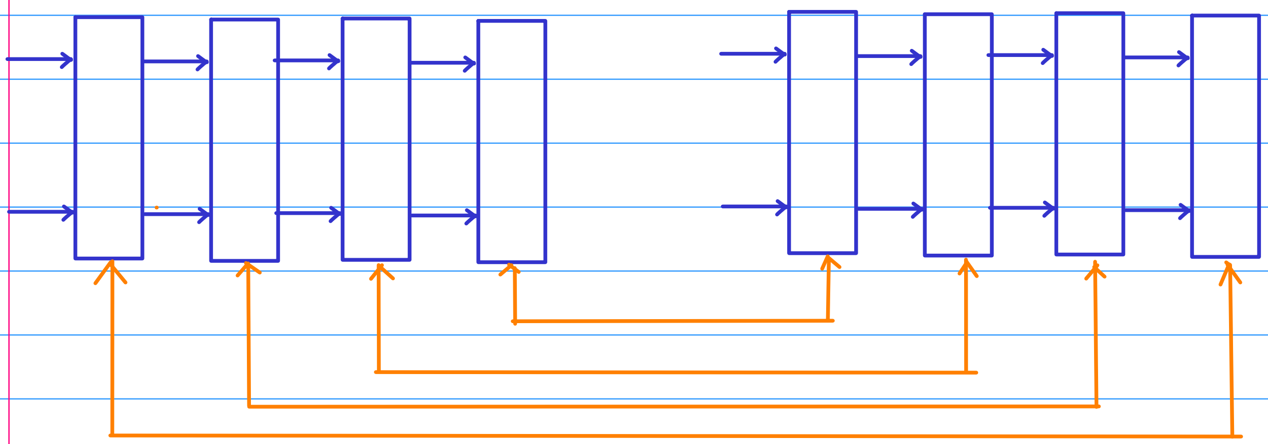
II → Baker's

$\lvert x_{i-1} \rvert$	$\lvert x_i \rvert$	x_i
0	0	-1
0	1	-1
1	0	1
1	1	1

* S. Wang , Swartzlander 1995

Merged CORDIC Algorithms

↳ approximation error prevents more steps in parallel.



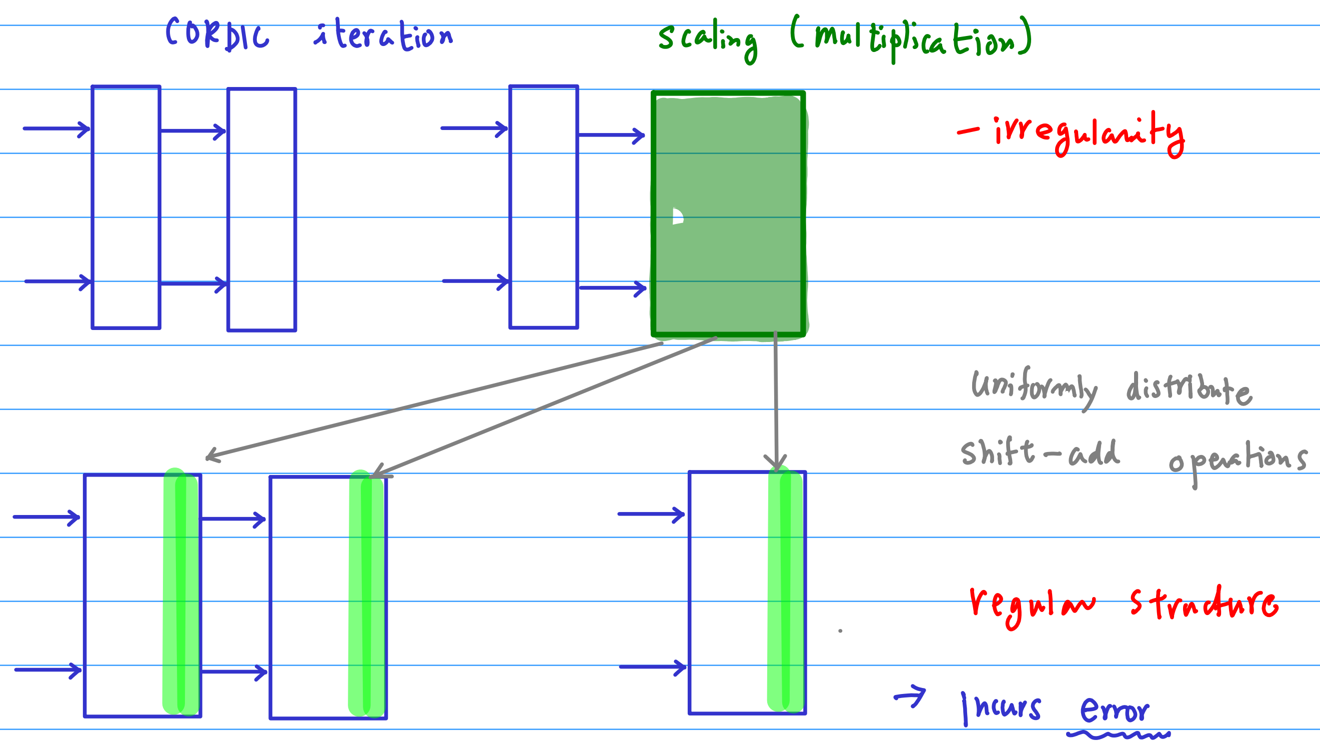
- rearrange the iteration sequence
- merge 2 iteration
- can reduce the size of barrel shifter by half.

$\frac{1}{2}$.

* S. Wang, Swartzlander 1997

Merged Scaling Multiplication CORDIC Algorithms

↳ approximation error prevents more steps in parallel.



the inspection of the p Most Significant Digits

$\sigma_i = 0$ avoided

the latency time of the inspection
increases with p
↳ fast carry-dependent adder

$p \leq 4 \sim 5$

allow $\sigma_i = 0$ valid choice

$$x[j+1] = x[j] + \sigma_j 2^{-j} y[j]$$
$$y[j+1] = y[j] - \sigma_j 2^{-j} x[j]$$

$$w[j] = 2^{j} y[j]$$
$$2^{-2j} w[j] = 2^{-j} y[j]$$

$$x[j+1] = x[j] + 2^{-2j} w[j]$$
$$y[j+1] =$$

the Original iteration process
which is necessary for finding proper σ_i's

⇒ recording the binary representation of z_i's
in the Signed Digit (SD) notation of σ_i's

Simple Bit Conversion Rule

can directly obtain $\boxed{\text{all } \sigma_i}$ from $\boxed{z_0}$
can be carried out in parallel
significant speed improvement

z_{i+1}	z_i	σ_i
0	0	-1
0	1	-1
1	0	1
1	1	1

2 D

Binary 0 0 1 0 1 1 0 1

1 -1 -1 1 -1 1 1 -1

$1\ \bar{1}\ \bar{1}\ 1\ \bar{1}\ 1\ 1\ \bar{1}$

the lsb : +1 (positive number)
-1 (negative number)

$2 \times 16 + 13$
$= 32$
13
45

1 T T 1 T 1 1 T

Lookahead

$$+\tan^{-1}\left(\frac{1}{2^0}\right)$$
$$-\tan^{-1}\left(\frac{1}{2^1}\right)$$
$$-\tan^{-1}\left(\frac{1}{2^2}\right)$$
$$+\tan^{-1}\left(\frac{1}{2^3}\right)$$
$$-\tan^{-1}\left(\frac{1}{2^4}\right)$$
$$+\tan^{-1}\left(\frac{1}{2^5}\right)$$
$$+\tan^{-1}\left(\frac{1}{2^6}\right)$$
$$-\tan^{-1}\left(\frac{1}{2^7}\right)$$

```
(%i1) a(i) := atan(1/2^i);
```
(%o1) $a(i):=\mathrm{atan}\left(\frac{1}{2^i}\right)$

```
(%i2) a(0);
```
(%o2) $\frac{\pi}{4}$

```
(%i4) %theta : a(0) - a(1) - a(2) + a(3) - a(4) + a(5) + a(6) - a(7);
```
(%o4) $-\mathrm{atan}\left(\frac{1}{2}\right)-\mathrm{atan}\left(\frac{1}{4}\right)+\mathrm{atan}\left(\frac{1}{8}\right)-\mathrm{atan}\left(\frac{1}{16}\right)+\mathrm{atan}\left(\frac{1}{32}\right)+\mathrm{atan}\left(\frac{1}{64}\right)-\mathrm{atan}\left(\frac{1}{128}\right)+\frac{\pi}{4}$

```
(%i5) float(%theta);
(%o5) 0.17775929681122
```

Lookahead

circular, hyperbolic coordinate system ($m \neq 0$)

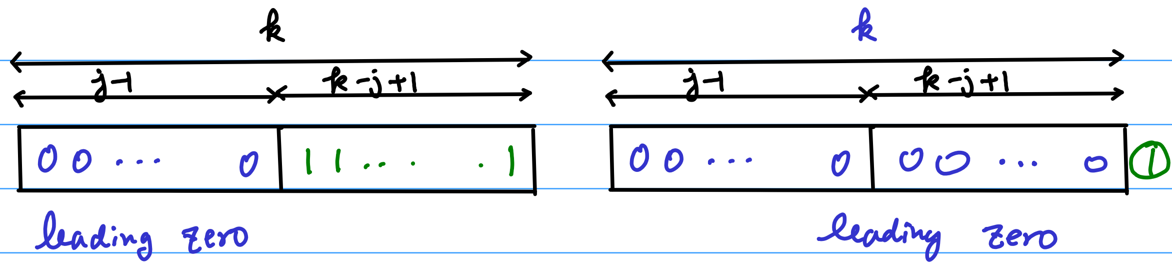
complex conversion $\leftarrow$ prediction error corrected.

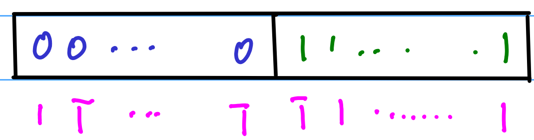
direct conversion not possible

$$\alpha_{m,i} \neq 2^{-i}$$

But $\alpha_{m,i} \approx 2^{-i}$ for sufficiently large i

can find the upper bound for the prediction error.





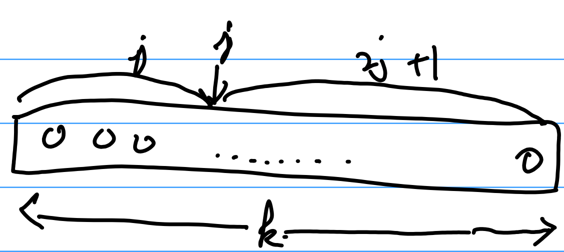
$$\boxed{\sum_{i=j}^{k} \left| 2^{-k} - \frac{1}{\sqrt{m}} \tan^{-1}\left(\sqrt{m}\, 2^{-s(m,i)}\right) \right| \le 2^{-s(m,k)}}$$

$s(m,i) = i, \qquad m = \pm 1$

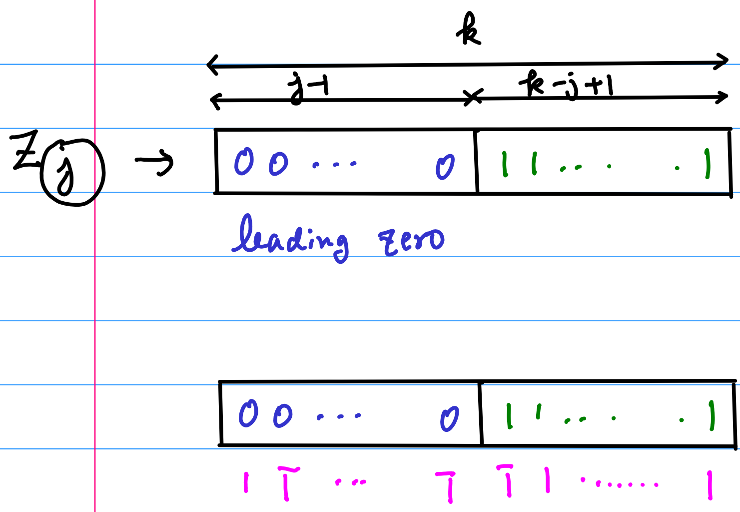
$j > 0$

$$\boxed{k \le 3j+1}$$

j	max k	$k \le 3j+1$
0	1	$= 3 \cdot 0 + 1$
1	4	$= 3 \cdot 1 + 1$
4	13	$= 3 \cdot 4 + 1$
13	40	$= 3 \cdot 13 + 1$
40	121	$= 3 \cdot 40 + 1$



Lookahead



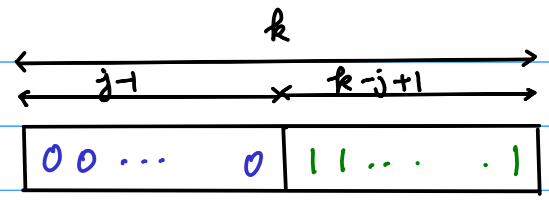
j-th iteration

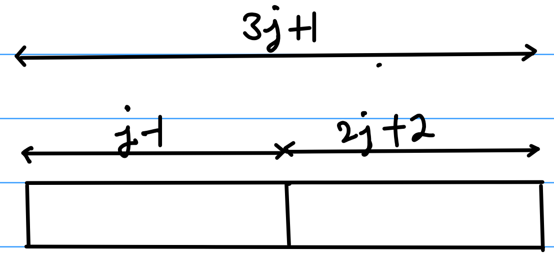
$Z_{(j)}$ assumed to have $(j-1)$ leading zero followed by $(k-j+1)$ ones

apply bit conversion rule

z_{i+1}	z_i	σ_i
0	0	-1
0	1	-1
1	0	1
1	1	1

Lookahead





Starting from the j-th iteration
the fully parallel bit conversion rule
for determining σ_i
may be applied
for the next $2j+2$ iterations
iteration $j \sim 3j+1$

followed by a repetition of the last iteration
in order to correct any possible errors.

* Timmerman 1992

Low Latency Time CORDIC Algorithms.

redundant addition

booth encoding

$\sigma_i = \pm 1$, when $\boxed{\sigma_i = 0}$ stop, freeze iteration → affects scale.

affects k_m, making it data-dependant.

parallelizing in determining σ_i

prior knowledge of σ_i

2 basic rotations in parallel
in 2 separate modules

2 module perform distinct computation
only when the algorithm "branches"

2 circular rotations in a single step
each step involves distinct computation → better.

* Baker's prediction scheme

P.W. Baker Suggestion for a fast binary sine/cosine generator.

$$(1 + j E_k 2^{-k})$$

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

$$\sin X = Im(U_n)$$
$$\cos X = Re(U_n)$$

$$U_{k+1} = U_k (1 + j E_k 2^{-k})$$

scale. $\beta = \prod_{k=0}^{n/2} (1 + 2^{-k})^{-1/2}$

$0 \leq k \leq n$ n: the bit length of X

when E_k's can be predicted

since $\tan^{-1}(2^{-k}) = 2^{-k} - \frac{2^{-3k}}{3} + \dots$

```
(%i21) taylor(atan(x), x, 0, 16);
```

(%o21)/T/ $x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \frac{x^9}{9} - \frac{x^{11}}{11} + \frac{x^{13}}{13} - \frac{x^{15}}{15} + \dots$

$\rightarrow$ given X_l with $(l-1)$ leading zeros

$$X_\ell = \pm 0.00 \cdots\cdots 0\, x_{\ell+1}\, x_{\ell+2} \cdots x_{2\ell} \cdots x_{3\ell}\, x_{3\ell+1} \cdots$$

by setting $x_i = E_i \qquad \ell \le i \le 3\ell$

$X_0 = X, \qquad U_0 = \beta + j\,0 \qquad \beta = 0.60 \cdots\cdot$

$X_{k+1} = X_k - E_k T_k \qquad X_{k+1} \to 0 \qquad 0 \le k \le n$

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

guaranteed $\quad X_{3\ell+1} \to (3\ell-1)$ leading zeros

the 3ℓ-th bit $\quad X_{3\ell+1} \to +1\ (-1)$

when the 3rd order term of $\tan^{-1}(2^{-k})$

produces carry (borrow)

out of the $(3\ell+1)$st bit position

into the (3ℓ)-th bit position

$\lvert x_{i-1} \rvert$	$\lvert x_i \rvert$	$\dot{x}_i$
0	0	-1
0	1	-1
1	0	1
1	1	1

(2) Phatak, D.S. (1998)

Double Step Branching CORDIC

IEEE Trans. on Computers, 47, 589-602

2 rotations are executed in a single step

more complicated ω-data path

several most significant digits are examined

Duprat & Muller 1993

CORDIC, New Results Fast VLSI Implementation.

Comments on Duprat and Muller's

Branching CORDIC

(3) Kwak, Choi, Swartzlander 2000

High speed CORDIC based on the overlapped architecture

Journal of VLSI Signal Processing 25, 167–178

1st rotation directions are predicted based on the approximation of the binary angle input

* Application Specific Processor

edited by Swartzlander

→ books.google.com

Ch2. Modelling the power consumption

Ch3. Fault tolerant Arithmetic

ch4. Low Power Digital Multiplier

Ch5. Unified View of CORDIC processor design

Ch6. Multi-dimensional Systolic Arrays → Hyesook Lim.

↳ DCT, DFT.