Redundant CORDIC
Ercegovac

20160723

Copyright (c) 2015 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".

.

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)

Ercegovac : Redundant & On-line CORDIC

① Modification of the standard CORDIC module for $\tan^{-1}(a/b) = \theta_i$ → redundant addition
→ variable scaling factor

② the angle transmitted in decomposed form to the rotation unit
→ reduces the communication bandwidth
→ eliminates the angle recurrance
$\boxed{\text{EUD}}$ [10]

③ the rotation module → using on-line addition [11, 18]

replacing the area consuming shifters with the area efficient delays

[10] Deprettere, Dewilde, Udo, "Pipelined CORDIC architecture for fast filtering and array processing
[11] Ercegovac, "On-line arithmetic: An Overview"
[18] Irwin, Owens, "Digit-pipelined arithmeticas ... paste-up system

[2] - fp CORDIC
H. M. Ahmed, Signal Processing Algorithms and Architectures
Dissertation Stanford, 1982

good for floating point

$$\theta = \tan^{-1}\left(\frac{a}{b}\right)$$

$$x_a[j+1] = x_a[j] + \sigma_j \cdot 2^{-j}\, y_a[j]$$
$$y_a[j+1] = y_a[j] - \sigma_j \cdot 2^{-j}\, x_a[j]$$
$$z_a[j+1] = z_a[j] + \sigma_j \cdot \tan^{-1}(2^{-j})$$

$$x_a[0] = b \qquad x_a[n] = K(a^2+b^2)^{1/2}$$
$$y_a[0] = a \qquad y_a[n] =$$
$$z_a[0] = 0 \qquad z_a[n] = 0$$

$$K = \prod_{j=0}^{n-1} (1 + \sigma_j^2\, 2^{-2j})^{1/2}$$

$$\sigma_j = \begin{cases} 1 & \text{if } y_a[j] \geq 0 \\ -1 & \text{if } y_a[j] < 0 \end{cases}$$

$$x_a[j+1] = x_a[j] + \sigma_j \boxed{2^{-j} y_a[j]}$$
$$\left(y_a[j+1] = y_a[j] - \sigma_j \boxed{2^{-j} x_a[j]} \right)$$
$$z_a[j+1] = z_a[j] + \sigma_j \tan^{-1}(2^{-j})$$

① elimination of one shifter

$$\boxed{2^{-j} y_a[j]} = 2^{-2j} \boxed{2^{+j} y_a[j]} = 2^{-2j} w[j]$$

$$x_a[j+1] = x_a[j] + \sigma_j 2^{-2j} w[j]$$

$$w[j] = \boxed{2^{+j} y_a[j]}$$
$$w[j+1] = 2^{+j+1} \left(y_a[j+1] \right)$$
$$= 2^{+j+1} \left(y_a[j] - \sigma_j \boxed{2^{-j} x_a[j]} \right)$$
$$= 2 \boxed{2^{+j} y_a[j]} - 2\sigma_j x_a[j]$$
$$= 2 \left(w[j] - \sigma_j x_a[j] \right)$$

$$w[j+1] = 2 \left(w[j] - \sigma_j x_a[j] \right)$$

$$x_a[j+1] = x_a[j] + \sigma_j 2^{-2j} w[j]$$

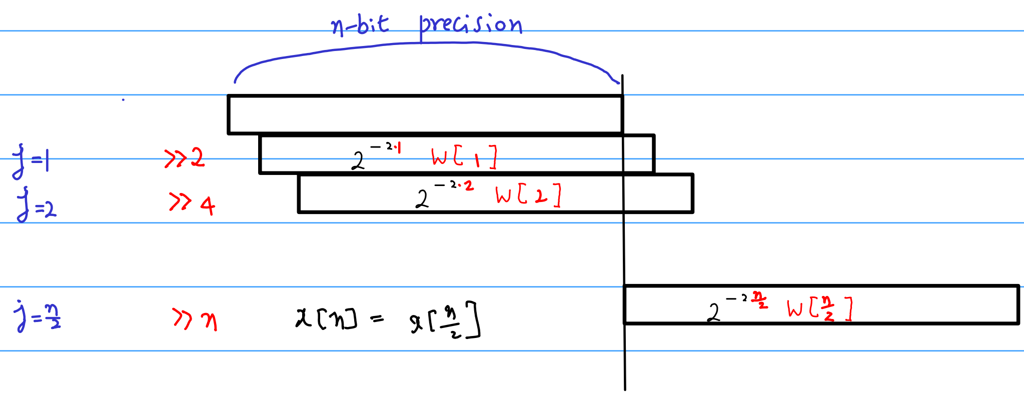
$$w[j+1] = 2(w[j] - \sigma_j x_a[j])$$

$$z_a[j+1] = z_a[j] + \sigma_j \tan^{-1}(2^{-j})$$

$$\sigma_j = \begin{cases} 1 & \text{if } w[j] \geq 0 \\ -1 & \text{if } w[j] < 0 \end{cases}$$

One shifter

$$x_a[j+1] = x_a[j] + \sigma \cdot \boxed{2^{-2j}} w[j]$$



$$x[0] \to x[1] \to x[2] \to \cdots \to x[\tfrac{n}{2}] = x[\tfrac{n}{2}+1] = \cdots = x[n]$$

(2) replacing the carry propagation addition by a redundant addition

- carry save
- signed digit

requires that the determination of σ_j uses an estimates of $w[j]$

→ necessary to produce a redundant representation of θ in terms of θ_i's

→ use $\{-1, 0, +1\}$

$$\boxed{\sigma_j = \begin{cases} 1 & \text{if} \quad w[j] \geq 0 \\ -1 & \text{if} \quad w[j] < 0 \end{cases}} \Rightarrow$$

* Carry-Save selection function

$$\boxed{\sigma_j = \begin{cases} +1 & \text{if} \quad \hat{w}[j] \geq 0 \\ 0 & \text{if} \quad \hat{w}[j] = -\frac{1}{2} \\ -1 & \text{if} \quad \hat{w}[j] \leq -1 \end{cases}}$$

$\hat{w}[j]$ an estimate of $w[j]$
with precision of 1 fractional bit

* Signed-digit selection function

$$\boxed{\sigma_j = \begin{cases} +1 & \text{if} \quad \hat{w}[j] \geq \frac{1}{2} \\ 0 & \text{if} \quad \hat{w}[j] = 0 \\ -1 & \text{if} \quad \hat{w}[j] \leq -\frac{1}{2} \end{cases}}$$

$\hat{w}[j]$ an estimate of $w[j]$
with precision of 1 fractional bit

The resulting θ. can be computed by 2 methods

① the angle in decomposed form
- by sequence of σ_j's

- the angle recurrence z_i not necessary

- directly used in the rotation for the triangularization calculation of θ_r & θ_l of SVD

② the angle is represented by $\boxed{\text{carry}}$ and $\boxed{\text{save}}$

On-line implementation

- redundant parallel adder
- on-line adder

the recurrence is unfolded and on-line adder is used

area consuming shifter → more efficient delays redundant number

σ_i must be estimated

all the inspected digits are zero

the proper value of σ_i

cannot be determined

without the knowledge of the remaining digit

assign 0 to σ_i freezing iteration

increasing CORDIC speed

CORDIC + redundant adder

parallelizing the determination of σ_i