CORDIC - Basic Algorithm and Enhancements

K.Sridharan, IIT Madras

February 25, 2010

The CORDIC algorithm provides an iterative method of performing vector rotations by arbitrary angles using only shifts and adds.

Vector rotation transform: For rotating in a Cartesian plane by angle ϕ.

$$\begin{aligned} x' &= x\cos\phi - y\sin\phi \\ y' &= y\cos\phi + x\sin\phi \end{aligned}$$

The CORDIC algorithm provides an iterative method of performing vector rotations by arbitrary angles using only shifts and adds.

Vector rotation transform: For rotating in a Cartesian plane by angle ϕ.

$$\begin{aligned} x' &= x \cos\phi - y \sin\phi \\ y' &= y \cos\phi + x \sin\phi \end{aligned}$$

OR

$$\begin{aligned} x' &= \cos\phi[x - y \tan\phi] \\ y' &= \cos\phi[y + x \tan\phi] \end{aligned}$$

If rotation angles are selected such that $\tan\phi = \pm 2^{-i}$, then

$$\begin{aligned} X_{i+1} &= K_i(X_i - Y_i \cdot d_i \cdot 2^{-i}) \\ Y_{i+1} &= K_i(Y_i + X_i \cdot d_i \cdot 2^{-i}) \\ Z_{i+1} &= Z_i - d_i \cdot \tan^{-1}(2^{-i}) \end{aligned}$$

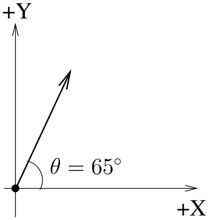
where

$$K_i = \cos(\tan^{-1}(2^{-i})) = \frac{1}{\sqrt{1+2^{-2i}}} \tag{1}$$

The scale factor K_i can be accumulated and the vector is scaled by

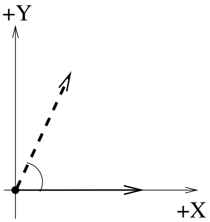
$$A_n = \prod_n \sqrt{1+2^{-2i}}$$

Polar (R, θ) to Rectangular (X,Y) transformation:



Rotation by 65°

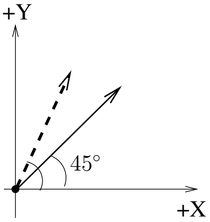
Polar (R, θ) to Rectangular (X,Y) transformation:



Initialize

$$X_0 = R$$
$$Y_0 = 0$$
$$Z_0 = \theta$$

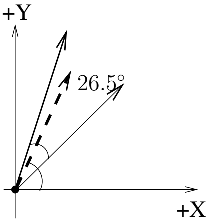
Polar (R, θ) to Rectangular (X,Y) transformation:



Rotate by $\tan^{-1}(2^0) = 45^\circ$

$$X_1 = X_0 - Y_0$$
$$Y_1 = Y_0 + X_0$$
$$Z_1 = Z_0 - \tan^{-1}(2^0)$$

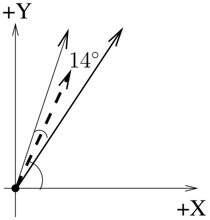
Polar (R, θ) to Rectangular (X,Y) transformation:



Rotate by
$\tan^{-1}(2^{-1}) = 26.5°$

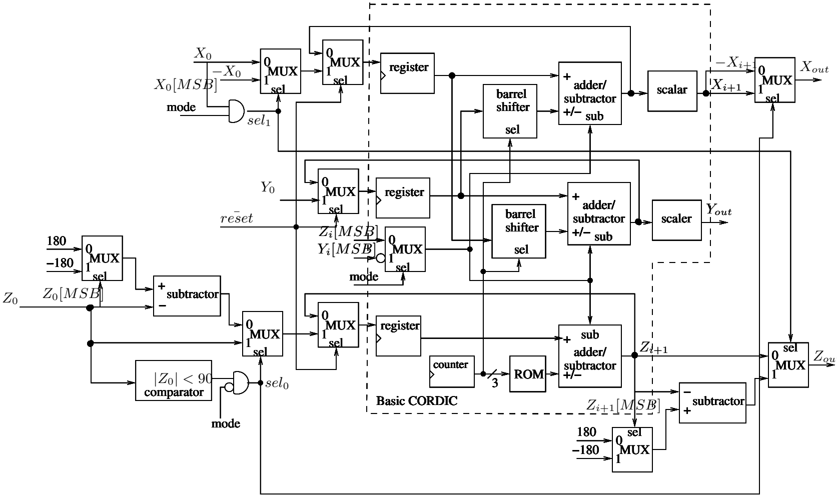
$$X_2 = X_1 - \frac{Y_1}{2}$$
$$Y_2 = Y_1 + \frac{X_1}{2}$$
$$Z_2 = Z_1 - \tan^{-1}(2^{-1})$$

Polar (R, θ) to Rectangular (X,Y) transformation:

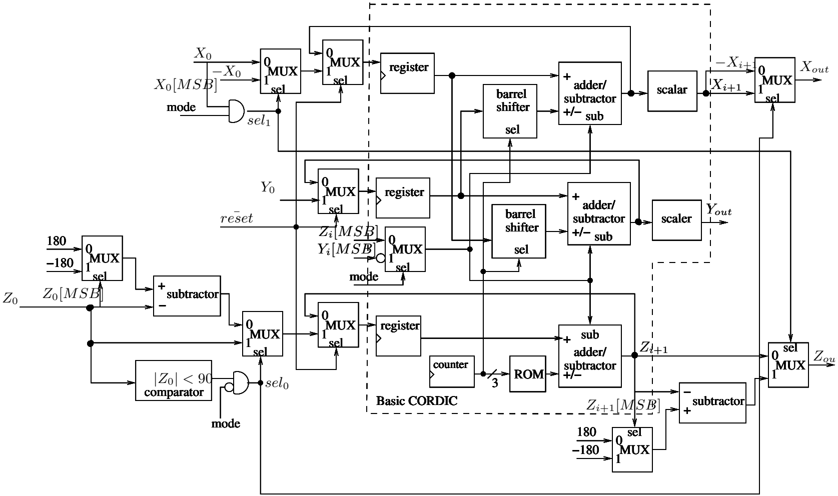


Rotate by $\tan^{-1}(2^{-2}) = 14^{\circ}$

$$
\begin{aligned}
X_3 &= X_2 + \frac{Y_2}{4} \\
Y_3 &= Y_2 - \frac{X_2}{4} \\
Z_3 &= Z_2 + \tan^{-1}(2^{-2})
\end{aligned}
$$



X_0
$-X_0$
$X_0[MSB]$
mode
sel_1
MUX
sel
register
barrel shifter
adder/ subtractor
+/− sub
scalar
$-X_{i+1}$
X_{i+1}
X_{out}
Y_0
$\bar{reset}$
$Z_i[MSB]$
$Y_i[MSB]$
mode
scaler
Y_{out}
180
−180
Z_0
$Z_0[MSB]$
subtractor
$|Z_0| < 90$
comparator
sel_0
mode
counter
3
ROM
Basic CORDIC
Z_{i+1}
$Z_{i+1}[MSB]$
Z_{ou}



X0
−X0
X0[MSB]
mode
sel1
MUX
sel
register
adder/
subtractor
+/− sub
barrel
shifter
scalar
−Xi+1
Xi+1
Xout
Y0
reset
Zi[MSB]
Yi[MSB]
mode
scaler
Yout
180
−180
Z0
Z0[MSB]
subtractor
|Z0| < 90
comparator
sel0
mode
counter
3
ROM
Basic CORDIC
Zi+1
Zi+1[MSB]
Zou

Reduce area consumption without affecting the performance in terms of **accuracy** and **number of iterations**.

- ROM : The size of the ROM is $2^{\lceil \log_2(no.\ of\ iterations) \rceil}$.
- Barrel shifters.
- Range is limited to $|Z| <= 99^\circ$. Multiplexers (both at input and output) are required to extend the range.

1. Completely eliminates barrel-shifters.
2. Represents all the angles in $[-180^\circ, 180^\circ]$ using combinations of two signed elementary angles, $tan^{-1}2^{-1}$ and $tan^{-1}2^{-3}$.

OR

$$Z = k_0 \cdot \tan^{-1}(2^{-1}) + k_1 \tan^{-1}(2^{-3})$$

- Either

$$
\begin{aligned}
X &= K_1 \cdot (X - (-1)^{sgn(k_0)} \cdot Y.2^{-1}) \\
Y &= K_1 \cdot (Y + (-1)^{sgn(k_0)} \cdot X.2^{-1})
\end{aligned}
$$

- Or

$$X = X - (-1)^{sgn(k_1)} \cdot Y \cdot 2^{-3}$$
$$Y = Y + (-1)^{sgn(k_1)} \cdot X \cdot 2^{-3}$$

- Either

$$\begin{aligned} X &= K_1 \cdot (X - (-1)^{sgn(k_0)} \cdot Y.2^{-1}) \\ Y &= K_1 \cdot (Y + (-1)^{sgn(k_0)} \cdot X.2^{-1}) \end{aligned}$$

- Or

$$X = X - (-1)^{sgn(k_1)} \cdot Y \cdot 2^{-3}$$
$$Y = Y + (-1)^{sgn(k_1)} \cdot X \cdot 2^{-3}$$

$$\max(|k_0| + |k_1|) = 13$$

for

$$170^\circ = 4 \cdot \tan^{-1}(2^{-1}) + 9 \cdot \tan^{-1}(2^{-3})$$

and

$$177^\circ = 8 \cdot \tan^{-1}(2^{-1}) - 5 \cdot \tan^{-1}(2^{-3})$$

Found using C program.

$$\max(|k_0| + |k_1|) = 13$$

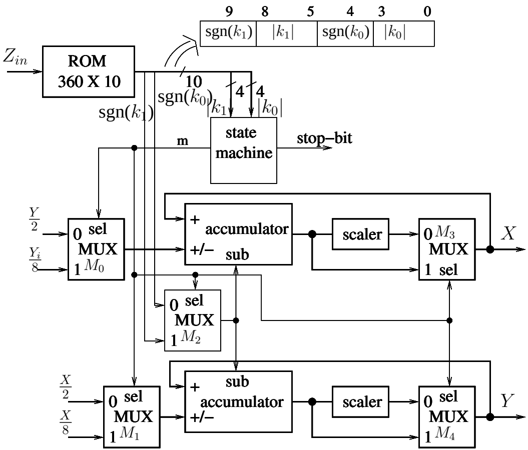
for

$$170^\circ = 4 \cdot \tan^{-1}(2^{-1}) + 9 \cdot \tan^{-1}(2^{-3})$$

and

$$177^\circ = 8 \cdot \tan^{-1}(2^{-1}) - 5 \cdot \tan^{-1}(2^{-3})$$

Found using C program.


9 8 5 4 3 0
sgn(k_1)
$|k_1|$
sgn(k_0)
$|k_0|$
Z_{in}
ROM
360 X 10
10
sgn(k_0)
$|k_1|$
4
4
$|k_0|$
sgn(k_1)
m
state machine
stop–bit
$\frac{Y}{2}$
$\frac{Y_i}{8}$
0 sel MUX 1 M_0
+ accumulator
+/− sub
scaler
0 M_3 MUX 1 sel
X
0 sel MUX 1 M_2
$\frac{X}{2}$
$\frac{X}{8}$
0 sel MUX 1 M_1
sub
+ accumulator
+/−
scaler
0 sel MUX 1 M_4
Y

1. J. Volder, The CORDIC trigonometric computing technique, IRE Transactions on Electronic Computers, Vol. EC-8, 1959, pp. 330-334.
2. J.S. Walther, A unified algorithm for elementary functions, Proceedings of 38th Spring Joint Computer Conference, 1971, pp. 379-385
3. R. Andraka, A survey of CORDIC algorithms for FPGA-based computers, Proceedings of ACM/SIGDA Sixth International Symposium on Field Programmable Gate Arrays, 1998, pp. 191-200
4. Leena Vachhani, K. Sridharan, P.K. Meher, Efficient CORDIC algorithms and architectures for low area and high throughput implementation, IEEE Transactions on Circuits and Systems - Part II:Express Briefs, Vol. 56, No. 1, January 2009, pp. 61-65