Chapter 1

CORDIC

1.1 Algoritmo CORDIC

El algoritmo CORDIC propuesto por Volder realizaba rotaciones en coordenadas circulares. Partiendo de esa base es facil extender su funcionamiento para que realice rotaciones en coordenadas hiperbólicas y lineales. Para lograrlo se agrega una variables que modifica las ecuaciones y además se eligen diferentes angulos para el acumulador de angule (variable z del algoritmo). Las ecuaciones del CORDIC completo son:

$$\begin{aligned} x_{n+1} &= x_n - m \cdot d_n \cdot 2^{-s_{m,n}} \\ y_{n+1} &= y_n + d_n \cdot 2^{-s_{m,n}} \\ z_{n+1} &= z_n - d_n \cdot \alpha_{m,n} \end{aligned} \tag{1.2}$$

Donde N representa la cantidad de pasos del algoritmo y se cumple que $n = 0, 1, 2, ..., N-1$. $s_{m,n}$ es una sequencia de números enteros no decreciente llamada *shift sequence* y $\alpha_{m,n}$ representa los angulos rotados para las diferentes coordenadas. d_n es una variable de control que maneja los sumadores/restadores.

$$\alpha_{m,n} = \frac{1}{\sqrt{m}} \cdot \tan^{-1}(\sqrt{m} \cdot 2^{-s_{m,n}}) \tag{1.3}$$

Eligiendo correctamente d_n y $s_{m,n}$ el algoritmo converge (ver tabla 1.1). La tabla 1.2 muestra las diferentes salidas del algoritmo. En la table **??** podemos ver las salidas del algoritmo para valores de entrada particulares, estos valores son de especial interés ya que representan operaciones matematicas difíciles de calcular.

Sistema de coordenadas	Shift sequence	Convergencia	Factor de escala
m	$s_{m,n}$	$\|A_0\|$	$K_m(n \to \inf)$
1	0,1,2, ... , n	1.74	1.64676
0	1,2,3, ... , n+1	1.0	1.0
-1	1,2,3,4,4, ...	1.13	0.82816

Table 1.1: CORDIC shift sequences

m	Modo	Entradas	Salidas
1	rotation	$x_0 = x$	$x_N = K_1 \cdot (x\cos\theta - y\sin\theta)$
		$y_0 = y$	$y_N = K_1 \cdot (y\cos\theta + x\sin\theta)$
		$z_0 = \theta$	$z_N = 0$
1	vectoring	$x_0 = x$	$x_N = K_1 \cdot sign(x) \cdot \sqrt{x^2 + y^2}$
		$y_0 = y$	$y_N = 0$
		$z_0 = \theta$	$z_N = \theta + \tan^{-1}(y/x)$
0	rotation	$x_0 = x$	$x_N = x$
		$y_0 = y$	$y_N = y + x \cdot z$
		$z_0 = \theta$	$z_N = 0$
0	vectoring	$x_0 = x$	$x_N = x$
		$y_0 = y$	$y_N = 0$
		$z_0 = z$	$z_N = z + y/x$
-1	rotation	$x_0 = x$	$x_N = K_{-1} \cdot (x\cosh\theta + y\sinh\theta)$
		$y_0 = y$	$y_N = K_{-1} \cdot (y\cosh\theta + x\sinh\theta)$
		$z_0 = \theta$	$z_N = 0$
-1	vectoring	$x_0 = x$	$x_N = K_{-1} \cdot sign(x) \cdot \sqrt{x^2 - y^2}$
		$y_0 = y$	$y_N = 0$
		$z_0 = \theta$	$z_N = \theta + \tanh^{-1}(y/x)$

Table 1.2: Salidas del algoritmo CORDIC.

Chapter 2

Algoritmo de Briggs para ln(x)

Si se encuentra una secuencia d_k tal que la productoria de x con $(1 + d_k 2^{-k})$ es cercana a 1 entonces vale que:

$$\begin{aligned} x \prod_{k=1}^{n} (1 + d_k 2^{-k}) &\approx 1 \\ ln(x) &\approx -\sum_{k=1}^{n} \ln(1 + d_k 2^{-k}) \end{aligned} \tag{2.2}$$

Chapter 3

BKM

3.1 Origenes

Consideremos el paso básico del algoritmo CORDIC en modo trigonométrico (con m=1). Si definimos el número complejo $E_n = x_n + j\, y_n$ con $j = \sqrt{-1}$, obtenemos $E_{n+1} = E_n(1 + j\, d_n 2^{-n})$, esta relación es similar al paso básico del algorithmo de Briggs. Esta similitud nos lleva a una generalización de ese algoritmo: podriamos realizar multiplicaciones por terminos $(1 + d_n 2^{-n})$, donde los d_ns son números complejos elejidos de tal manera que la multiplicación por d_n pueda reducirce a unas pocas sumas. Entonces se define el algoritmo BKM de la siguiente manera:

$$\begin{cases} E_{n+1} = E_n \cdot (1 + d_n 2^{-n}) \\ L_{n+1} = L_n - \ln(1 + d_n 2^{-n}) \end{cases} \tag{3.1}$$

con $d_n \in \{0, \pm 1, \pm j, \pm 1 \pm j\}$ y $\ln z$ es el número complejo t tal que $\exp t = z$ y cuya parte imaginaria está entre $-\pi$ y π.

3.2 E-mode

Encontrar una sequencia de d_n tal que $L_n \to 0$ entonces $E_n \to E_1 e^{L_1}$.

$$\begin{cases} E_n \to E_1 e^{L_1} \\ L_n \to 0 \end{cases} \tag{3.2}$$

3.2.1 Canal de datos

$$\begin{cases} E^x_{n+1} = E^x_n + (d^x_n En^x - d^y_n E^y_n)\, 2^{-n} \\ E^y_{n+1} = E^y_n + (d^x_n En^y + d^y_n E^x_n)\, 2^{-n} \end{cases} \tag{3.3}$$

3.2.2 Canal de control

$$\begin{cases} \quad l_n = 2^n L_n \\ l_{n+1} = 2l_n - 2^{n+1}\ln(1 + d_n 2^{-n}) \\ l^x_{n+1} = 2l^x_n - 2^n \ln[1 + d^x_n\, 2^{-n+1} - (d^{x\,2}_n + d^{y\,2}_n)\, 2^{-2n}] \\ l^y_{n+1} = 2l^y_n - 2^{n+1} d^y_n \arctan\left(\dfrac{2^{-n}}{1 + d^x_n 2^{-n}}\right) \end{cases} \tag{3.4}$$

$$2^n \ln(1 + d_n 2^{-n}) \to 1 \qquad as\ n \to +\infty \tag{3.5}$$

3.3 L-mode

Encontrar una sequencia de d_n tal que $E_n \to 1$ entonces $L_n \to L_1 + \ln(E_1)$.

$$\begin{cases} E_n \to 1 \\ L_n \to L_1 + \ln(E_1) \end{cases} \tag{3.6}$$

3.3.1 Canal de datos

$$\begin{cases} L^x_{n+1} = L^x_n - \dfrac{1}{2}\ln[1 + d^x_n\, 2^{-n+1} - (d^{x\,2}_n + d^{y\,2}_n)\, 2^{-2n}] \\ L^y_{n+1} = L^y_n - d^y_n \arctan\left(\dfrac{2^{-n}}{1 + d^x_n 2^{-n}}\right) \end{cases} \tag{3.7}$$

3.3.2 Canal de control

$$\begin{cases} \quad e_n = 2^n(E_n - 1) \\ e_{n+1} = 2(e_n + d_n) + d_n e_n 2^{-n+1} \\ e^x_{n+1} = 2(e^x_n + d^x_n) + (d^x_n e^x_n - d^y_n e^y_n)2^{-n+1} \\ e^y_{n+1} = 2(e^y_n + d^y_n) + (d^y_n e^x_n + d^x_n e^y_n)2^{-n+1} \end{cases} \tag{3.8}$$

Chapter 4

Architecture

4.1 BKM Float

4.2 BKM Fixed

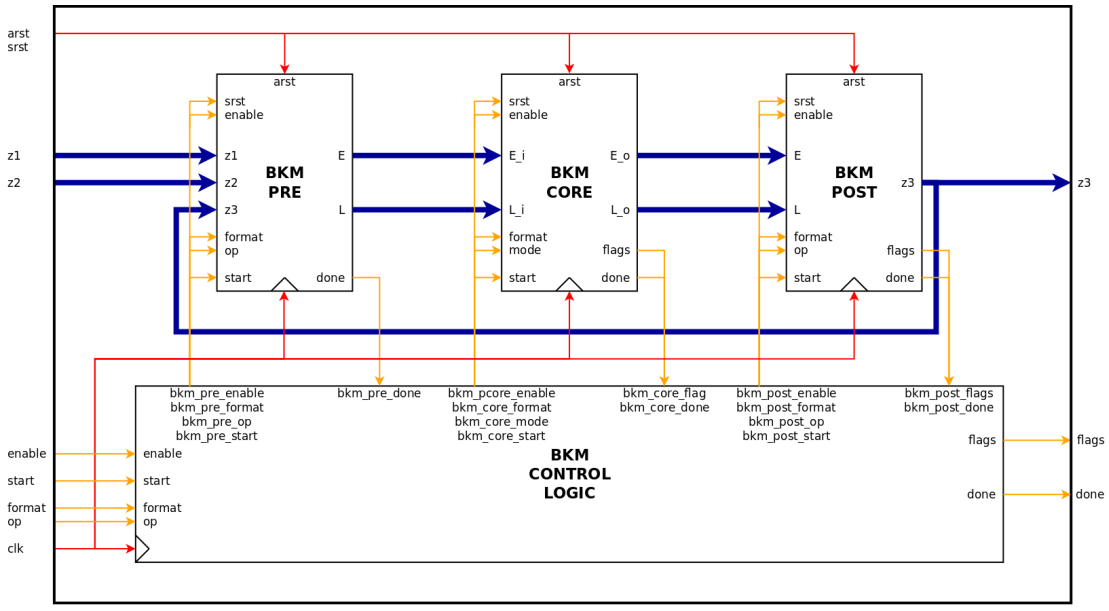

Figure 4.1: Arquitecture del bloque BKM fixed.

Puerto	Tamaño	Dirección	Descripción
clk	1 bit	Input	Clock signal
arst	1 bit	Input	Active high asynchronous reset signal
srst	1 bit	Input	Active high synchronous reset signal
enable	1 bit	Input	Active high enable signal
start	1 bit	Input	Active high start signal
format	2 bit	Input	Format specifier (0/1): MSB for complex/real, LSB for 64/32 bits
op	5 bits	Input	Operation code
x_1	64 bits	Input	Real part of input variable z1
y_1	64 bits	Input	Imaginary part of input variable z1
x_2	64 bits	Input	Real part of input variable z2
y_2	64 bits	Input	Imaginary part of input variable z2
x_3	64 bits	Output	Real part of output variable z3
y_3	64 bits	Output	Imaginary part of output variable z3
flags	1 bit	Output	Active high invert output signal
done	1 bit	Output	Active high done signal: Stays asserted until start strobe

Table 4.1: Puertos del bloque op decoder.

4.2.1 BKM Pre

4.2.2 BKM Post

4.2.3 BKM Fixed control logic

4.3 BKM Core