IP Core CORDIC

Trabajo Final de Microarquitecturas y Softcores

Carrera de Especialización en Sistemas Embebidos

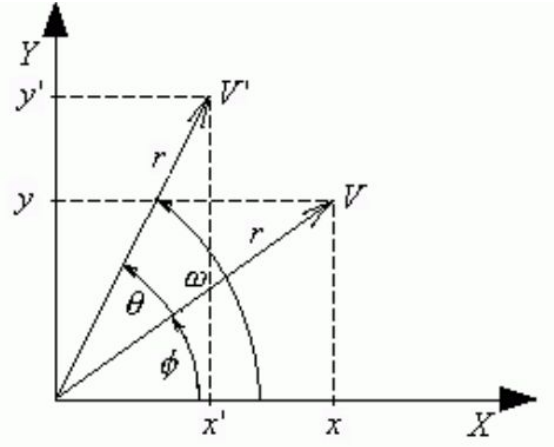
Autor: Ing. Lucas Pablo Meoli

Docente: Ing. Nicolás Álvarez

Introducción

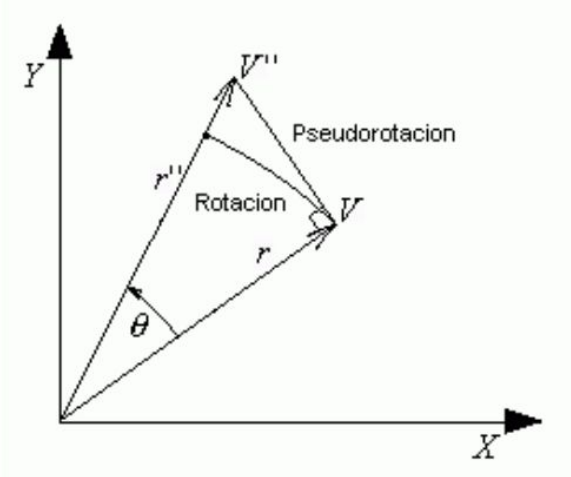
- CORDIC iterativo
- Diseño IP core
- Sistema base utilizando el IP CORDIC
- Prueba y validación del IP core

Funcionamiento



$$x' = x\cos\theta - y\,\mathrm{sen}\,\theta$$
$$y' = y\cos\theta + x\,\mathrm{sen}\,\theta$$

Rotación de un vector



$$x' = \cos\theta\left(x - y\,\mathrm{tg}\,\theta\right)$$
$$y' = \cos\theta\left(y + x\,\mathrm{tg}\,\theta\right)$$

Pseudorotación de un vector

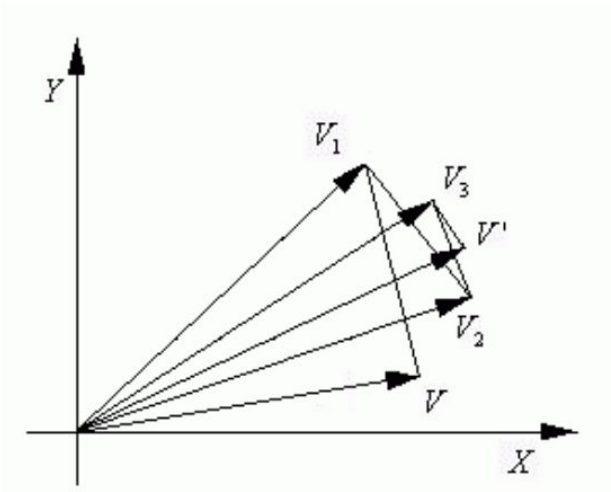
Funcionamiento

Restringiendo el valor de la tangente: $\operatorname{tg}\theta = \pm 2^{-i}, i \in \mathrm{N}$

$$x_{i+1} = K_i \left(x_i - y_i d_i 2^{-i} \right)$$
$$y_{i+1} = K_i \left(y_i + x_i d_i 2^{-i} \right)$$

$$K_i = \cos\theta_i = \cos\left(\operatorname{arctg} 2^{-i}\right) = \frac{1}{\sqrt{1+\operatorname{tg}^2\theta_i}} = \frac{1}{\sqrt{1+2^{-2i}}}$$

Ecuaciones generales



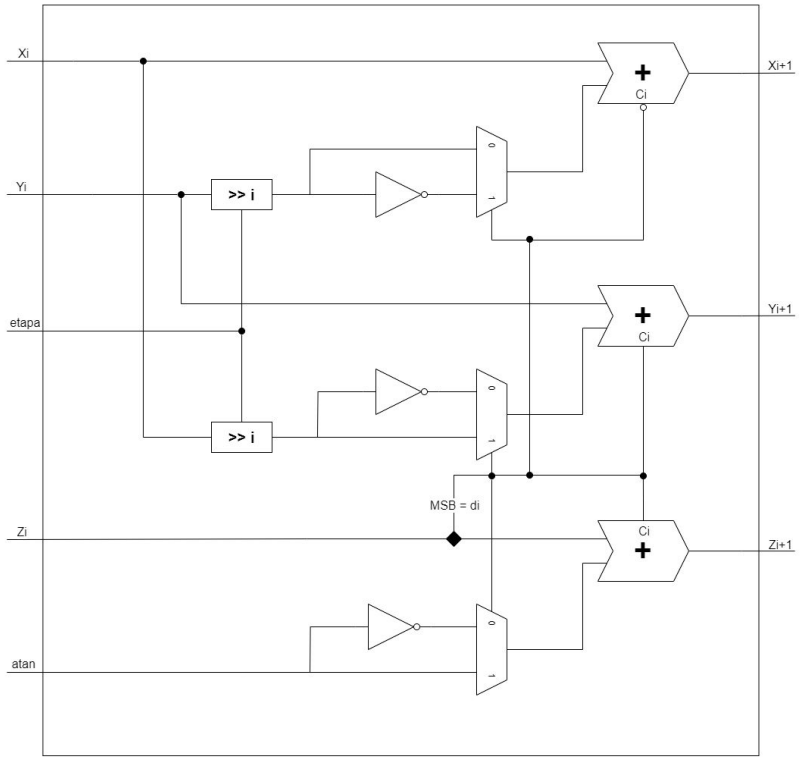
$$x_{i+1} = x_i - y_i d_i 2^{-i}$$
$$y_{i+1} = y_i + x_i d_i 2^{-i}$$
$$z_{i+1} = z_i - d_i \operatorname{arctg}\left(2^{-i}\right)$$

$$d_i = \begin{cases} -1 & , si\ z_i < 0 \\ 1 & , si\ z_i \geq 0 \end{cases}$$

Rotación por pseudorotaciones

Diseño

Módulo principal CORDIC



Recordatorio

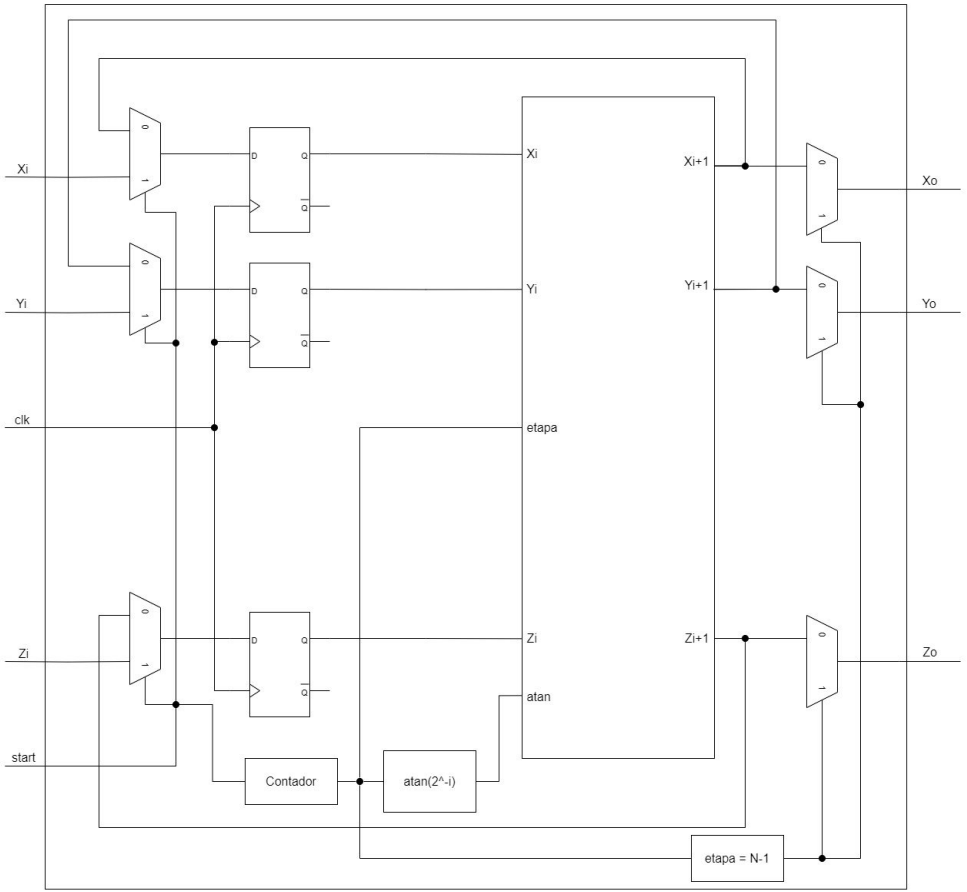
$$x_{i+1} = x_i - y_i d_i 2^{-i}$$

$$y_{i+1} = y_i + x_i d_i 2^{-i}$$

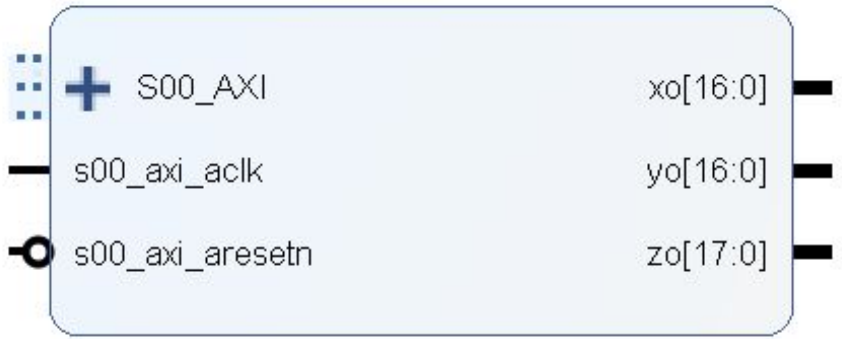
$$z_{i+1} = z_i - d_i \operatorname{arctg}\left(2^{-i}\right)$$

Diseño

Arquitectura iterativa



IP Core



- Interfaz AXI slave
- Utilización de los 4 registros
- 3 puertos de salida
- Parámetro para determinar la cantidad de bits

Sistema base

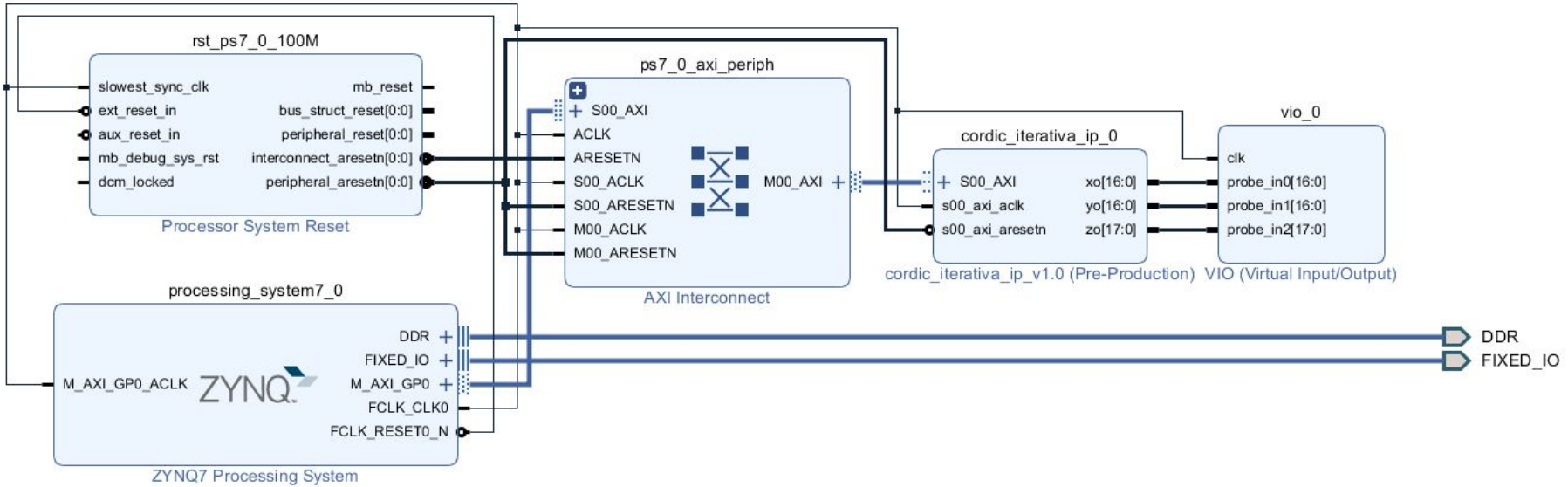


Tabla de recursos

Name	Slice LUTs (17600)	Slice Registers (35200)	Slice (4400)	LUT as Logic (17600)	LUT as Memory (6000)	LUT Flip Flop Pairs (17600)	Bonded IOB (100)	BUFGCTRL (32)	BSCANE2 (4)
cordic_VIO	6.14%	4.52%	10.41%	6.00%	0.40%	3.78%	1.00%	6.25%	25.00%
cordic_iterativa_inst (c...	1.37%	0.35%	1.68%	1.37%	0.00%	0.45%	0.00%	0.00%	0.00%
dbg_hub (dbg_hub)	2.63%	2.05%	5.14%	2.49%	0.40%	1.69%	0.00%	3.13%	25.00%
vio_inst (vio_0)	2.14%	2.12%	4.11%	2.14%	0.00%	1.59%	0.00%	0.00%	0.00%

Name	Slice LUTs (17600)	Slice Registers (35200)	Slice (4400)	LUT as Logic (17600)	LUT as Memory (6000)	LUT Flip Flop Pairs (17600)	Bonded IOPADs (130)	BUFGCTRL (32)	BSCANE2 (4)
system_wrapper	7.67%	5.67%	13.34%	7.19%	1.40%	4.83%	100.00%	6.25%	25.00%
dbg_hub (dbg_hub)	2.63%	2.05%	5.25%	2.49%	0.40%	1.75%	0.00%	3.13%	25.00%
inst (xsdbm_v3_0_0_xsdbm)	2.63%	2.05%	5.25%	2.49%	0.40%	1.75%	0.00%	3.13%	25.00%
system_i (system)	5.04%	3.62%	8.32%	4.70%	1.00%	3.02%	0.00%	3.13%	0.00%
cordic_iterativa_ip_0 (system_cordic_iterativa_ip_0_0)	1.55%	0.83%	2.20%	1.55%	0.00%	0.81%	0.00%	0.00%	0.00%
processing_system7_0 (system_processing_system7_0_0)	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	3.13%	0.00%
ps7_0_axi_periph (system_ps7_0_axi_periph_0)	1.92%	1.24%	2.98%	1.59%	0.98%	0.98%	0.00%	0.00%	0.00%
rst_ps7_0_100M (system_rst_ps7_0_100M_0)	0.09%	0.09%	0.25%	0.09%	0.02%	0.08%	0.00%	0.00%	0.00%
vio_0 (system_vio_0_0)	1.48%	1.46%	3.30%	1.48%	0.00%	1.15%	0.00%	0.00%	0.00%

Recorte aplicación en C

system.hdf | system.mss | cordic.c | cordic_iterativa_ip.h | xil_io.h

```c
#include "xparameters.h"
#include "xil_io.h"
#include "cordic_iterativa_ip.h"

//====================================================

int main (void)
{
   int i;

   xil_printf("-- Start of the Program --\r\n");

   while (1)
   {
      // output dip switches value on LED_ip device
      CORDIC_ITERATIVA_IP_mWriteReg(XPAR_CORDIC_ITERATIVA_IP_0_S00_AXI_BASEADDR, CORDIC_ITERATIVA_IP_S00_AXI_SLV_REG3_OFFSET, 0);
      CORDIC_ITERATIVA_IP_mWriteReg(XPAR_CORDIC_ITERATIVA_IP_0_S00_AXI_BASEADDR, CORDIC_ITERATIVA_IP_S00_AXI_SLV_REG0_OFFSET, 1000);
      CORDIC_ITERATIVA_IP_mWriteReg(XPAR_CORDIC_ITERATIVA_IP_0_S00_AXI_BASEADDR, CORDIC_ITERATIVA_IP_S00_AXI_SLV_REG1_OFFSET, 0);
      CORDIC_ITERATIVA_IP_mWriteReg(XPAR_CORDIC_ITERATIVA_IP_0_S00_AXI_BASEADDR, CORDIC_ITERATIVA_IP_S00_AXI_SLV_REG2_OFFSET, 32768);
      CORDIC_ITERATIVA_IP_mWriteReg(XPAR_CORDIC_ITERATIVA_IP_0_S00_AXI_BASEADDR, CORDIC_ITERATIVA_IP_S00_AXI_SLV_REG3_OFFSET, 1);
      for (i=0; i<9999999; i++);
      xil_printf("-- Loop end --\r\n");
   }
}
```

Prueba en FPGA

hw_vio_1

Name	Value	Activity	Direction	VIO
system_i/cordic_iterativa_ip_0_xo[16:0]	[U] 824		Input	hw_vio_1
system_i/cordic_iterativa_ip_0_yo[16:0]	[U] 1427		Input	hw_vio_1
system_i/cordic_iterativa_ip_0_zo[17:0]	[U] 0		Input	hw_vio_1

Rotación 60°

hw_vio_1

Name	Value	Activity	Direction	VIO
system_i/cordic_iterativa_ip_0_xo[16:0]	[U] 0		Input	hw_vio_1
system_i/cordic_iterativa_ip_0_yo[16:0]	[U] 2715		Input	hw_vio_1
system_i/cordic_iterativa_ip_0_zo[17:0]	[U] 0		Input	hw_vio_1

Rotación 30°

hw_vio_1

Name	Value	Activity	Direction	VIO
system_i/cordic_iterativa_ip_0_xo[16:0]	[U] 1165		Input	hw_vio_1
system_i/cordic_iterativa_ip_0_yo[16:0]	[U] 1164		Input	hw_vio_1
system_i/cordic_iterativa_ip_0_zo[17:0]	[U] 0		Input	hw_vio_1

Rotación 45°

hw_vio_1

Name	Value	Activity	Direction	VIO
system_i/cordic_iterativa_ip_0_xo[16:0]	[U] 0		Input	hw_vio_1
system_i/cordic_iterativa_ip_0_yo[16:0]	[U] 1649		Input	hw_vio_1
system_i/cordic_iterativa_ip_0_zo[17:0]	[U] 0		Input	hw_vio_1

Rotación 90°

Prueba en FPGA

hw_vio_1

Name	Value	Activity	Direction	VIO
system_i/cordic_iterativa_ip_0_xo[16:0]	[S] -1164		Input	hw_vio_1
system_i/cordic_iterativa_ip_0_yo[16:0]	[U] 1164		Input	hw_vio_1
system_i/cordic_iterativa_ip_0_zo[17:0]	[U] 0		Input	hw_vio_1

Rotación 135°

hw_vio_1

Name	Value	Activity	Direction	VIO
system_i/cordic_iterativa_ip_0_xo[16:0]	[S] 8232		Input	hw_vio_1
system_i/cordic_iterativa_ip_0_yo[16:0]	[S] 8235		Input	hw_vio_1
system_i/cordic_iterativa_ip_0_zo[17:0]	[U] 0		Input	hw_vio_1

Rotación 180°

¡Gracias!
¿Preguntas?