



Tecnura

ISSN: 0123-921X

tecnura.ud@correo.udistrital.edu.co

Universidad Distrital Francisco José de
Caldas
Colombia

Gómez, Edwar Jacinto; Platero Plazas, Donovan Camilo; Robayo Restrepo, Mario
Fernando

Voltmetro TRUE-RMS sobre FPGA basado en algoritmo cordic

Tecnura, vol. 19*, noviembre, 2015, pp. 129-136

Universidad Distrital Francisco José de Caldas
Bogotá, Colombia

Disponible en: <http://www.redalyc.org/articulo.oa?id=257059819016>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica

Red de Revistas Científicas de América Latina, el Caribe, España y Portugal

Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto



Voltímetro TRUE-RMS sobre FPGA basado en algoritmo cordic

TRUE-RMS voltmeter on FPGA cordic-based algorithm

Edwar Jacinto Gómez¹, Donovan Camilo Platero Plazas²,
Mario Fernando Robayo Restrepo³

Fecha de recepción: Agosto 28 de 2015

Fecha de aceptación: Septiembre 25 de 2015

Como citar: Jacinto, E., Platero, D., & Robayo, M. (2015). Voltímetro TRUE-RMS sobre FPGA basado en algoritmo CORDIC. Revista Tecnura, 19 (CITIE), 129-136. doi: <http://dx.doi.org/10.14483/udistrital.jour.tecnura.2015.ICE.a15>

Resumen

El cálculo del verdadero valor RMS de una señal de tensión, en corriente alterna, es un asunto de precisión y de accesibilidad. Primero: los resultados pueden alterarse dependiendo del tipo de onda, y segundo: cuando se requiera el procesamiento de operaciones complejas en un multímetro TRUE RMS, por ejemplo, basado en microcontroladores simples, se exige un elevado recurso computacional eventualmente costoso. El presente artículo, describe la investigación que condujo a la implementación de un Voltímetro true-RMS sobre FPGA SPARTAN 3E 1600, utilizando VHDL, que mide con plausible exactitud el valor eficaz de una señal, discretizada con conversor LTC-1407-1A, en circuitos de corriente continua y/o en corriente alterna. La innovación introducida radica en aplicar el algoritmo CORDIC (*COordinate Rotation Digital Computer*), desarrollado originalmente como una solución digital para los problemas de navegación en tiempo real sobre FPGA (*Field Programmable Gate Arrays*) para las operaciones matemáticas tales como la extracción

de raíces cuadradas, entre otras. Se obtiene así un desarrollo sobre un dispositivo digital que es computacionalmente eficiente.

Palabras claves: True-RMS, FPGA, CORDIC, Algoritmos de Aproximación, Raíz Cuadrada, Vectores.

Abstract

The Calculation of true-rms of an AC signal, it's an issue of accuracy and accessibility. First of all: the results can be upset depending of the kind of wave, and second of all: when it's required the complex operations processing on a true-rms multimeter, e.g., based on simple microcontrollers, this requires a high computation resource eventually expensive. This article, describes the research that it routed to the implementations of a true-rms voltmeter on SPARTAN 3E 1600 FPGA using VHDL; it measure with plausible accuracy, the true-rms of a discretized signal with LTC-1407-1A ADC converter, on AC or DC circuits. The innovation is located in apply the CORDIC algorithm (it was developed mainly to resolve navigation problems), FPGA-based to the math

¹ Magister en ciencias de la información y las comunicaciones, Universidad Distrital Francisco José de Caldas, Ingeniero en Control Electrónico e Instrumentación, Universidad Distrital Francisco José de Caldas, Tecnólogo en Electrónica, Universidad Distrital Francisco José de Caldas, Bogotá D.C., Colombia. Contacto: ejacintog@udistrital.edu.co

² Estudiante Ingeniería en Control, Universidad Distrital Francisco José de Caldas, Tecnólogo en Electrónica, Universidad Distrital Francisco José de Caldas, Bogotá D.C., Colombia. Contacto: dcplaterop@correo.udistrital.edu.co

³ Estudiante Ingeniería en Control, Universidad Distrital Francisco José de Caldas, Tecnólogo en Electrónica, Universidad Distrital Francisco José de Caldas, Bogotá D.C., Colombia. Contacto: mfroboyar@correo.udistrital.edu.co

operations such as the extraction of square roots, between others. In this way, accessibility and computational efficiency are obtained on the digital device.

Keywords: True-RMS, FPGA, CORDIC, Approximation algorithms, square root, vectors.

INTRODUCCIÓN

Actualmente se pueden encontrar instrumentos digitales que calculan el verdadero valor RMS de una señal de tensión en corriente alterna con gran precisión pero con la desventaja de que su costo es elevado.

Por ejemplo un multímetro de gama baja-media tiene la capacidad de medir tensiones continuas con una precisión del 10%, o menos; pero no mide el verdadero valor eficaz de una señal alterna debido a que siempre toma la señal de entrada como una onda sinusoidal, es decir, los resultados van a ser alterados dependiendo del tipo de onda (Pérez, 1999).

Por otra parte, cuando se va a diseñar una aplicación que requiera el procesamiento de operaciones más complejas; como ocurre con un multímetro TRUE RMS, es posible utilizar un microcontrolador para realizarlas, pero el resultado nunca será fiable debido a que en microcontroladores simples, operaciones, tales como la multiplicación, se vuelven una tarea tediosa, imprecisa, y muchas veces el hardware interno del microcontrolador no posee la robustez necesaria. Aproximadamente, el proceso de multiplicación puede durar de 3 a 40 ciclos de máquina del dispositivo; otras operaciones (raíces cuadradas, exponenciales, logaritmos naturales, funciones trigonométricas, funciones hiperbólicas, modulo y fase de un vector) exigen mayor tiempo y esto da a entender que el proceso de operaciones complejas en microcontroladores simples exige un recurso computacional muy elevado, reduciendo sensiblemente el rendimiento de tales aplicaciones.

Actualmente, aunque haya avanzado el hardware de tipo digital, hace falta implementar, herramientas matemáticas que eleven el rendimiento de la ejecución de operaciones como las mencionadas

anteriormente. Es por eso que se ha optado, para el desarrollo del instrumento, por el algoritmo denominado CORDIC, el cual permite realizar operaciones complejas con sumas y corrimientos de bits. En consecuencia se eleva la velocidad de procesamiento digital de la señal adquirida, logrando de paso que el dispositivo, además de eficiente, sea de bajo costo, comparado con algunas alternativas comerciales.

Aunque el algoritmo CORDIC se desarrolló hacia la década de los 60 del siglo pasado, sus aplicaciones son de este siglo. En 2006, en Francia, se desarrolló un sintetizador de frecuencia digital directa con algoritmo CORDIC y aproximación de series de Taylor para receptores digitales. Este combina dos aspectos teóricos interesantes en cuanto a Ingeniería. Primero, aplica las series de Taylor, para aproximar la función seno y mejorar la velocidad de los tiempos de acceso. Y en segundo lugar, la lógica digital propia del algoritmo CORDIC para la potencia, redundando en disminución del área hardware de la aplicación, (Jridi & Alfalou, 2009).

En 2008 el Grupo de Automática y Robótica, GAR, de la Universidad Javeriana, en Cali, Colombia; implementó la tangente inversa usando el algoritmo CORDIC. Este proyecto base en la implementación del algoritmo CORDIC para calcular la tangente inversa de un ángulo determinado, usando una FPGA. Concluyen, que se aumenta en un 29% la velocidad de dicho proceso, respecto al core o núcleo que tiene la empresa Xilinx®, ya incluido en su plataforma de programación (Amaya & Velazco, 2008).

Por otra parte, un proyecto realizado en el departamento de ingeniería electrónica de la Universidad Politécnica de Valencia (España) hacia 2010, en el cual se implementó un analizador de espectros que, mide módulo y fase de señales de hasta

30 Mhz con una resolución mejor que 10 Khz. Este tiene un rango dinámico de 74dBs y permite ampliar el espectro por factores 2 y 4; utilizando un dispositivo lógico programable (en este caso una FPGA Xtreme DSP Kit de Nallatech.). Los bloques constitutivos son: un bloque FFT (Fast Fourier Transform o Transformada Rápida de Fourier); bloque ROM; y el bloque CORDIC, que se utilizó para convertir la salida de la FFT a valores de módulo y fase; y para calcular el logaritmo y módulo de la señal para la representación en escala logarítmica (Sansaloni, Pérez, Torres, & Valls, n.d.). junto con muchos otros trabajos que buscan realizar calculos en punto fijo, tratando siempre de reducir la cantidad de operaciones necesarias para realizar un calculos matematicos (Hu & Naganathan, 1989) (Park & Cho, 2004) (Wu & Wu, 2001) (Gaj, n.d.) y ademas de esto su uso en diferentes tipos de aplicaciones, como las comunicaciones (Chen & Sima, 2011) (Cui, Yu, Sheng, & Cui, 2006), la robótica (Martínez, Acero & Castiblanco, 2015), o simplemente la realización de calculos matematicos simples (Lakshmi & Dhar, 2011) (Akhter, Fatema, Ferdouse, & Khandaker, 2010).

A continuación, se muestra el desarrollo para realizar este tipo de algoritmos sobre dispositivos digitales programables tipo FPGA y basados en

CORDIC; lo cual hace que la presente aplicación sea innovadora con respecto al uso del hardware y software embebido usado.

MATERIALES Y MÉTODOS

Diagrama de bloques de la solución

El diagrama de bloques de la solución se presenta en la figura 1.

En consecuencia de la metodología implica: acondicionar la señal; discretizarla; detectar su cruce por cero; controlar módulos; implementar CORDIC; almacenar variables; escribir datos y visualizarlos.

Acondicionamiento de la señal: Esta etapa se encarga del acondicionamiento de la señal de entrada, a través de una reducción de tensión mediante un divisor conformado por una serie de resistencias para que el ensión e entrada no supere el voltaje máximo admitido por el ADC.

Discretización de la Señal: En esta etapa se discretiza la señal utilizando un conversor análogo-digital LTC-1407-1A, el cual,permite que la señal analógica sea guardada como datos digitales para luego ser analizados y procesados hallando el valor rms de la misma.

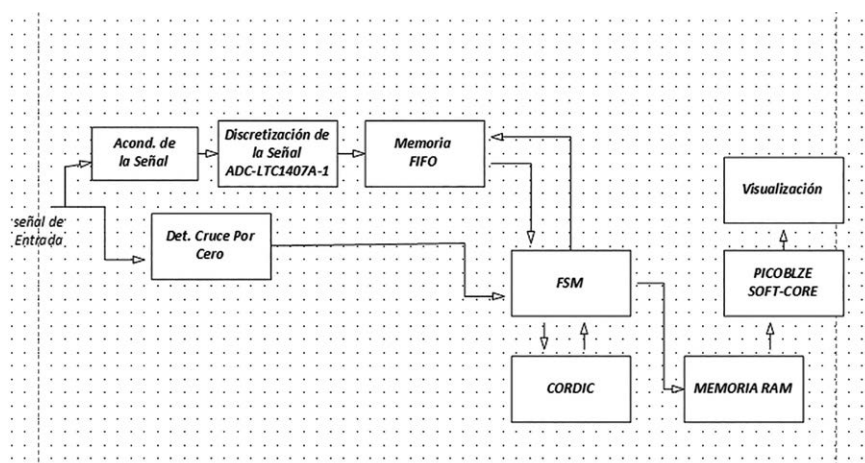


Figura 1. Diagrama de bloques del Voltímetro True-RMS.

Fuente: elaboración propia.

Detección de cruce por cero: Para los cálculos de periodo; valor que se necesita para hallar el V_{rms} , se utilizó un circuito de detección de cruce por cero. Este se compone de un comparador de tensión con este y un reloj rápido, generado en VHDL.

Máquina de Estados Finitos (FSM): Para el control de cada uno de los módulos en la aplicación, se definió una máquina de estados, implementada en VHDL, con un número de estados específicos, ya que se debe manejar el dispositivo de forma secuencial

Algoritmo CORDIC

El trabajo se centra en emplear CORDIC para la Raíz cuadrada. Modelo matemático para hallar el valor RMS de la señal.

CORDIC es un método iterativo para la rotación de ángulos específicos basado únicamente en sumas y corrimientos de bits (Liu & Feng, 2007; Maharatna, Troya, Krstic, Grass, & Jagdhold, 2004; Meher, Valls, Juang, Sridharan, & Maharatna, 2009; Yu & Swartzlander, 2002; Z Jin & Z Huajun, 2008). El algoritmo propuesto por Volder en 1959 consiste en la rotación de un vector bidimensional V en el plano, como se muestra en la figura 2.

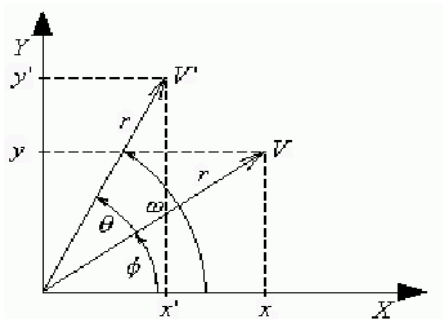


Figura 2. Rotación de un vector en un plano bidimensional.

Fuente: (Andraka, 1998; Volder, 1959; Walther, 2000).

De esta imagen se deducen las ecuaciones (1) y (2) para V' .

$$x' = x \cos \theta - y \sin \theta \quad (1)$$

$$y' = y \cos \theta + x \sin \theta \quad (2)$$

Factorizando se derivan las ecuaciones (3) y (4).

$$x' = \cos \theta (x - y \tan \theta) \quad (3)$$

$$y' = \cos \theta (y + x \tan \theta) \quad (4)$$

Si se hace $\tan \theta = 2^{-i}$ donde i es el número de la iteración actual desde 0 hasta N , las ecuaciones (3) y (4) pueden ser reescritas como se muestra en la ecuación (5), (6) y (7).

$$x_{i+1} = K_i (x_i - y_i d 2^{-i}) \quad (5)$$

$$y_{i+1} = K_i (y_i + x_i d 2^{-i}) \quad (6)$$

$$K_i = \frac{1}{\sqrt{1 + \tan^2 \theta}} = \frac{1}{\sqrt{1 + 2^{-2i}}}, d = \pm 1 \quad (7)$$

El modelo eficaz de la rotación, utilizado por el algoritmo CORDIC, se muestra en las ecuaciones (8) – (12).

$$x_{i+1} = x_i - y_i d (2^{-i}) \quad (8)$$

$$y_{i+1} = y_i + x_i d (2^{-i}) \quad (9)$$

$$z_{i+1} = z_i - d \arctan(2^{-i}) \quad (10)$$

$$d = -1 \text{ si } z_i < 0 \quad (11)$$

$$d = 1 \text{ si } z_i \geq 0; \quad (12)$$

Esta es la rotación circular que desarrolló Volder. Luego, más adelante, Walther en 1971 extendería completamente el algoritmo generalizándolo en las ecuaciones (13) – (15).

$$x_{i+1} = x_i - u y_i d (2^{-i}) \quad (13)$$

$$y_{i+1} = y_i + x_i d (2^{-i}) \quad (14)$$

$$z_{i+1} = z_i - d f (2^{-i}) \quad (15)$$

Dónde:

$u=1$ para rotación circular

$u=0$ para rotación lineal

$u=-1$ para rotación hiperbólica

Y

$f(x)=\arctan(x)$ para rotación circular

$f(x)=x$ para rotación lineal

$f(x)=\operatorname{arctanh}(x)$ para rotación hiperbólica

Es en el caso de la rotación hiperbólica en donde se encuentra la solución a la raíz cuadrada de un número (Edwards, 2000), a partir de las ecuaciones (13) – (15), de la siguiente manera:

$$d = -1 \text{ si } y_i \geq 0 \quad (16)$$

$$d = 1 \text{ si } y_i < 0; \quad (17)$$

Esta rotación produce las ecuaciones (18) y (20).

$$x_n = A_n \sqrt{x_0^2 - y_0^2} \quad (18)$$

$$y_n = 0 \quad (19)$$

$$z_n = z_0 + \log_2^{-1} \left(\frac{y_0}{x_0} \right) \quad (20)$$

Con $A_n \cong 0.80$.

Si al comenzar el algoritmo, los valores iniciales de x_0 y y_0 son tales que:

$$x_0 = u + \frac{1}{4} \quad (21)$$

$$y_0 = u - \frac{1}{4} \quad (22)$$

Se obtendrá finalmente la raíz cuadrada x_n del número u . [15-20]

$$x_n = \sqrt{\left(u + \frac{1}{4}\right)^2 - \left(u - \frac{1}{4}\right)^2} \quad (23)$$

$$x_n = \sqrt{u^2 + \frac{1}{2}u + \frac{1}{16} - u^2 + \frac{1}{2}u - \frac{1}{16}} \quad (24)$$

$$x_n = A_n \sqrt{u} \quad (25)$$

Almacenamiento de Variables y Visualización

Para el almacenamiento de una memoria RAM donde se guardan los datos de las variables en posiciones específicas para que luego sea recorrida por el procesador Picoblaze y puedan ser mostradas en la LCD. El Picoblaze es un procesador embebido, de la empresa Xilinx. Este se encarga de escribir los datos procesados en la LCD. Finalmente, la visualización de las constantes calculadas se hace a través de un display LCD de 2 X 16 posiciones. Para desarrollar e integrar el dispositivo se utilizó una FPGA SPARTAN 3E 1600® la cual trae incluido el conversor análogo digital y la LCD.

Pruebas y resultados

Aunque no se puede hacer una comparación con dispositivos similares, las pruebas se realizaron con voltímetros basados en microcontroladores simples. Para ello se contrasta la medición con la realizada con un multímetro FLUKE87® y el dispositivo propuesto, variando la frecuencia de la señal medida entre 0 y 20.000Hz; el tipo de esta (Seno, Triangular, Cuadrada o DC) y el voltaje pico (entre 5 y 10V).

Los resultados se muestran en la tabla 1.

Se puede inferir que el error promedio de la FPGA propuesta es aproximado al 3% frente al 6% del FLUKE87.

El error del dispositivo propuesto es menor a mas altas frecuencias (20.000 Hz) y en señales triangulares; en tanto que las desviaciones mas altas ocurren a frecuencias medias de 4529 Hz y en señales seno, en ambos casos es mejor que en el FLUKE87.

CONCLUSIONES

Cuando se implementa un sistema de procesamiento digital de señal se logra obtener una mayor eficiencia y exactitud que en sistemas análogos (Hosticka, n.d.). Dicho esto se puede afirmar que se obtuvo un dispositivo para instrumentación

Tabla 1. Comparación entre Multímetro Fluke 87 y el voltímetro implementado.

No_dato	Frecuencia (Hz)	Tipo de Señal	Tensión pico	Tensión medido FPGA (V)	Tensión medido fluke 87 V(V)	Vrms real (V)	Error FPGA	Error Fluke
0	3061	seno	10	7,3	6,75	7,071067812	3,24%	5%
1	3061	triangular	10	5,976	5,76	5,773502692	3,51%	0%
2	3061	cuadrada	10	10,12	9,6	10	1,20%	4%
3	0	dc	10	10,23	9,8	10	2,30%	2%
4	60	seno	10	7,27	6,75	7,071067812	2,81%	5%
5	60	triangular	10	5,93	5,7	5,773502692	2,71%	1%
6	60	cuadrada	10	10,12	9,6	10	1,20%	4%
7	0	dc	10	10,223	9,8	10	2,23%	2%
8	60	seno	5	3,632	3,4	3,535533906	2,73%	4%
9	60	triangular	5	2,95	2,76	2,886751346	2,19%	4%
10	60	cuadrada	5	5,3	4,8	5	6,00%	4%
11	0	dc	5	5,2	4,98	5	4,00%	0%
12	4549	seno	5	3,29	3,1	3,535533906	6,94%	12%
13	4529	seno	10	7,31	6,68	7,071067812	3,38%	6%
14	4529	triangular	5	3,024	2,59	2,886751346	4,75%	10%
15	1000	cuadrada	5	5,017	4,557	5	0,34%	9%
16	4529	cuadrada	5	5,1	4,27	5	2,00%	15%
17	12930	seno	6,24	4,197	3,978	4,412346315	4,88%	10%
18	12930	triangular	6,24	3,39	3,28	3,60266568	5,90%	9%
19	12930	cuadrada	6,24	6,1	5,38	6,24	2,24%	14%
20	20000	seno	7,2	5,04	4,963	5,091168825	1,01%	3%
21	20000	triangular	7,2	4,178	4,089	4,156921938	0,51%	2%
22	20000	cuadrada	7,2	7,03	6,57	7,2	2,36%	9%

Fuente: elaboración propia.

análogo digital que puede ser utilizado en cualquier tipo de sistema electrónico, el cual analiza señales de tensión, tanto en corriente continua como en corriente alterna, para cualquier tipo de aplicación.

Un dispositivo con excelente precisión fabricado en Colombia, especial para aplicaciones de análisis y procesamiento de señal digital poniendo en práctica una tecnología que prácticamente no se ha utilizado en nuestro país y que se puede implementar en cualquier tipo de dispositivos digitales por el poco recurso computacional que exige y además por las operaciones complejas que es capaz de realizar.

Se logra implementar un voltímetro sobre FPGA SPARTAN 3E 1600, que permite calcular el valor

RMS con un porcentaje de error de aproximadamente 3%, a señales de corriente alterna que no superen los 150Vp y los 25kHz.

El uso de algoritmos computacionales para el procesamiento digital de señales es una alternativa que viable en sistemas digitales simples para realizar tareas complejas que no se podían realizar a priori; cálculos de senos, cosenos, logaritmos, raíces cuadradas, entre otros.

Las ventajas de los sistemas que usan técnicas de procesamiento digital, radican en que se basan en procesadores programables los cuales permiten mejorar continuamente las funcionalidades del mismo.

En todo tipo de sistemas de medida, se deben tener protecciones para los dispositivos utilizados

para que no sean afectados por una indebida manipulación; las alternativas digitales, en todo caso, son más seguras y permiten su fácil reparación.

Las implementaciones futuras de CORDIC, son múltiples, en particular en el procesamiento digital de señales y en presencia de operaciones complejas y recurrentes. Un estado de arte sistemático orientaría en esta vía.

RECONOCIMIENTOS

Reconocemos a los grupos de investigación DIGITI y SciBas, de sistemas digitales inteligentes y ciencias básicas respectivamente, adscritos al centro de investigaciones y desarrollo científico de la Universidad Distrital Francisco José de Caldas –CIDC–, por la asistencia teórica y práctica para la consecución del diseño e implementación propuesto.

FINANCIAMIENTO

Universidad Distrital Francisco José de Caldas

REFERENCIAS

- Akhter, N., Fatema, K., Ferdouse, L., & Khandaker, F. (2010). Implementation of the Trigonometric LMS Algorithm using Original Cordic Rotation. *International Journal of Computer Networks Communications*, 2(4), 12. Retrieved from <http://arxiv.org/abs/1008.3328>
- Amaya, F., & Velazco, J. (2008). Diseño de la tangente inversa usando el algoritmo cordic.
- Andraka, R. (1998). A survey of CORDIC algorithms for FPGA based computers. *Proceedings of the 1998 ACM/SIGDA Sixth International Symposium on Field Programmable Gate Arrays-FPGA '98*, 191–200. <http://doi.org/10.1145/275107.275139>
- Chen, D., & Sima, M. Fixed-Point CORDIC-Based QR Decomposition by Givens Rotations on FPGA, 2011 International Conference on Reconfigurable Computing and FPGAs 327–332 (2011). IEEE. <http://doi.org/10.1109/ReConFig.2011.38>
- Cui, X., Yu, D., Sheng, S., & Cui, X. (2006). A CORDIC Demodulator Platform for Digital-IF Receiver. *Computer*, 2–4.
- Edwards, B. (2000). Como multiplican y dividen las calculadoras CORDIC. *Enseñanza Universitaria*, 159–172.
- Gaj, K. (n.d.). Sine / Cosine using CORDIC Algorithm.
- Hosticka, B. J. (n.d.). Performance Comparison of MOS/LSI Analog and Digital Signal Processors.
- Hu, Y. H., & Naganathan, S. Angle recording method for efficient implementation of the CORDIC algorithm, 42 IEEE International Symposium on Circuits and Systems 99–102 (1989). Ieee. <http://doi.org/10.1109/ISCAS.1989.100320>
- Jridi, M., & Alfalou, A. (2009). Direct Digital Frequency Synthesizer with CORDIC Algorithm and Taylor Series Approximation for Digital Receivers. *European Journal of Scientific Research*, 30(4), 542–553.
- Lakshmi, B., & Dhar, A. S. Parallel CORDIC-like architecture: For fast rotation implementation, TENCON 2011 2011 IEEE Region 10 Conference 701–705 (2011). IEEE. Retrieved from <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6129199>
- Liu, G., & Feng, Q. (2007). ASIC design of low-power reconfigurable FFT processor. *2007 7th International Conference on ASIC*, 44–47. <http://doi.org/10.1109/ICASIC.2007.4415563>
- Maharatna, K., Troya, A., Krstic, M., Grass, E., & Jagdhold, U. (2004). A CORDIC like processor for computation of arctangent and absolute magnitude of a vector. *Architecture*, 713–716. Retrieved from <http://eprints.soton.ac.uk/263544/>
- Martínez, F.H., Acero, D., & Castiblanco, M. (2015). Robótica Autónoma: Acercamientos a algunos problemas centrales. Publicaciones Universidad Distrital Francisco José de Caldas.
- Meher, P., Valls, J., Juang, T.-B., Sridharan, K., & Maharatna, K. (2009). 50 Years of CORDIC: Algorithms, Architectures, and Applications. *IEEE Transactions on Circuits and Systems I Regular Papers*, 56(9), 1893–1907. Retrieved from <http://eprints.ecs.soton.ac.uk/17873/>

- Park, S. Y. P. S. Y., & Cho, N. I. C. N. I. Fixed-point error analysis of CORDIC processor based on the variance propagation formula, 51 IEEE Transactions on Circuits and Systems I Regular Papers 573–584 (2004). <http://doi.org/10.1109/TCSI.2003.820232>
- Pérez, P. F. (1999). Dispositivos de Medición. In *Curso de Capacitación en Medidas Electrónicas* (pp. 1–9).
- Sansaloni, T., Pérez, A., Torres, V., & Valls, J. (n.d.). Diseño de un analizador de espectros en FPGA, 6–8.
- Volder, J. E. (1959). The CORDIC Trigonometric Computing Technique. *Ieee Transactions On Electronic Computers, EC-8*(3), 330–334. <http://doi.org/10.1109/TEC.1959.5222693>
- Walther, J. S. (2000). The story of unified CORDIC. *The Journal of VLSI Signal Processing, 25*(2), 107–112.
- Retrieved from <http://www.springerlink.com/index/K21104P108676P0K.pdf>
- Wu, C.-S. W. C.-S., & Wu, A.-Y. W. A.-Y. Modified vector rotational CORDIC (MVR-CORDIC) algorithm and architecture, 48 Ieee Transactions On Circuits And Systems Ii Analog And Digital Signal Processing 548–561 (2001). <http://doi.org/10.1109/82.943326>
- Yu, S. Y. S., & Swartzlander, E. E. J. A scaled DCT architecture with the CORDIC algorithm, 50 IEEE Transactions on Signal Processing 160–167 (2002). <http://doi.org/10.1109/78.972492>
- Z Jin, & Z Huajun. (2008). Direct Exponential Function Computer in Neural Network. *Information Technology Journal, 7*, 1156–1162.

