ISSN (Print) : 0974-6846
**Indian Journal of Science and Technology,** *Vol 10(24), DOI: 10.17485/ijst/2017/v10i24/110790, June 2017* ISSN (Online) : 0974-5645

# A CORDIC based Configurable Fixed-Point Design on FPGA using Minimal Hardware

**G. Edwar Jacinto, Fredy H. Martínez* and S. Fernando Martínez**

Technological Faculty, District University Francisco José de Caldas, Bogotá D.C., Colombia;
ejacintog@udistrital.edu.co, fhmartinezs@udistrital.edu.co, fmartinezs@udistrital.edu.co

## Abstract

**Objectives:** To design a COordinate Rotation DIgital Computer (CORDIC) based electrical signal processing system for efficient and minimalist electrical power processing. **Methods:** This stage of assessment describes the design and performance of a true RMS (Root-Mean Square) voltage meter on Xilinx SPARTAN 3E 1600 FPGA. **Findings:** The effectiveness of the proposed designs is assessed through FPGA implementations and error simulations. Measurement results show that the model can reproduce behaviors similar to the original model traditional use of multipliers, but with less resource consumption (hardware and processing time). **Novelty:** The use of CORDIC to reduce the computational cost of the algorithm, and its implementation in an embedded system.

**Keywords:** CORDIC, Embedded System, FPGA, Real-Time, True RMS

## 1. Introduction

CORDIC is an iterative algorithm for calculating hyperbolic and trigonometric functions[1], and of course, for computing inverse trigonometric functions[2]. It is particularly suited to hardware implementations because it does not require any multiplies. Its importance is high in embedded systems because most processing algorithms require such operation[3]. The algorithm was originally developed as a digital solution for real-time navigation[4,5]. The algorithm allows to develop embedded systems for specific applications, in particular with low complexity and high accuracy characteristics, key elements on hardware implementation[6–8].

There are two modes of CORDIC: rotation and vectoring[9,10]. The vector is rotated by a desired angle in rotation mode, whereas the vector is aligned with X-axis in vectoring mode. In this research we use the rotation mode for the synthesis of functions. Fixed-angle-rotation operation is widely used in signal processing[11]. Various CORDIC designs have been proposed for uniform rotation of vectors through specified angles with modest use of resources (Look-Up Tables or LUTs, and Flip-Flops)[2,12,13]. In many cases the algorithms use pipeline to allow different functional units to operate concurrently[14].

Thinking about simplifying the implementation hardware, CORDIC is an iterative algorithm which requires shift and addition operations. One of the main uses of the algorithm is the hardware realization of the sine and cosine of an angle. From these functions it is possible to determine operations like multiplication, division, hyperbolic, exponential, logarithm and square root[5]. Thanks to the Field-Programmable Gate Array (FPGA) is a high-speed programmable device, it has been widely used to realize CORDIC based algorithms[15]. Their joint use has been particularly important in embedded systems and robotics, applications which guarantees an efficient system for calculating angles and distances[6].

The research group aims to develop low-cost, high-performance embedded devices for energy quality measurement, and active power factor correctors. In particular, this paper describes and shows the results of the design and operation of a true RMS voltage meter. The solution makes use of the CORDIC algorithm on an FPGA in order to increase performance and reduce costs.

The following part of the paper is arranged in this way. Section 2 presents preliminary concepts and problem formulation. Section 3 illustrates the methodology based

* *Author for correspondence*

on CORDIC algorithm for calculating the true RMS. Section 4 we present the preliminary results. And finally, in Section 5, we present our conclusions.

## 2. Problem Formulation

Periodic waves, such as the case of grid voltage $V(t)$, have frequency components that are multiple integers of some fundamental frequency. This voltage signal can be expressed as the sum of the fundamental, its harmonics, and some DC value ($V_0$, equation 1).

$$V(t) = V_0 + \sum_{h=1}^{N} V_h \sin[h\omega t + \theta_h] \quad (1)$$

Equation 2 shows how to find the RMS value of a voltage waveform, where the RMS value of each of the harmonics, $V_h$, is known.

$$V_{RMS} = \left[\sum_{h=1}^{N} (V_h)^2\right]^{\frac{1}{2}} \quad (2)$$

Many voltage measuring devices assume sinusoidal waveform, and calculate the value from the peak of the signal. Others calculate an average value. In both cases the reading is only correct if the signal is sine, and is erroneous in the cases of square, triangular, or periodic waves with harmonic content (general case of the power grid). Therefore, the calculation from equation 2 is called true RMS, and is expected in a good measurement equipment.

It is proposed to perform the square root calculation and the exponentiation, and even the approximation of the signal sampled between sample and sample, from the calculation of sines and cosines using the CORDIC algorithm. To determine the sine or cosine of an angle it is necessary to find the $x$ and $y$ coordinates of its vector in the unit circle Figure 1. CORDIC begins with vector $V_0$. In the first iteration, this vector is rotated 45 degrees counter-clockwise to get the vector $V_1$. Successive iterations rotate the vector in one or the other direction by size-decreasing steps until it reaches the desired angle. Every iteration calculates the rotation multiplying the vector $n$ with the rotation matrix $R_i$ (equation 3).

$$V_n = R_i * V_{i-n} \quad (3)$$

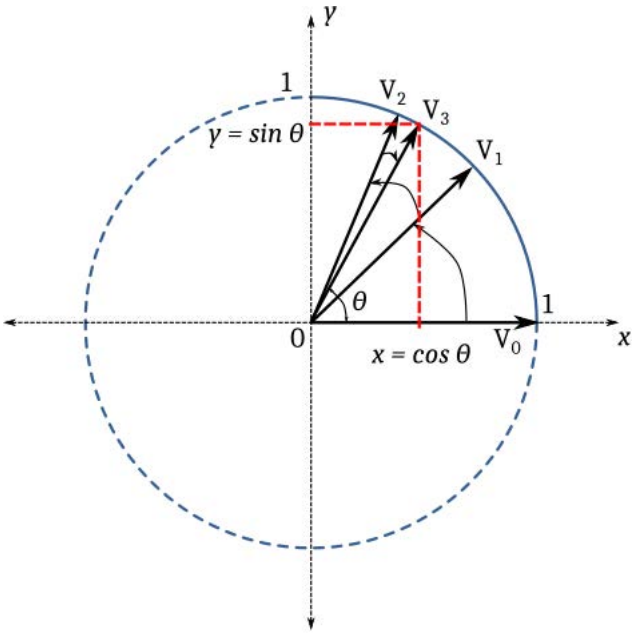


**Figure 1.** CORDIC vector rotation.

Where the rotation matrix is given by (equation 4):

$$R_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{bmatrix} \quad (4)$$

The direction of rotation is defined according to the error with respect to the desired vector. Knowing further that (equation 5):

$$\cos\theta = \frac{1}{\sqrt{1+tan^2\theta}}, \sin\theta = \frac{\tan\theta}{\sqrt{1+tan^2\theta}} \quad (5)$$

After some iterations the angle of the vector will get close enough to $\theta$.

Our research seeks to implement this strategy for the development of embedded systems dedicated to the real time processing of electric power quality. For this, we evaluate the performance of three CORDIC cores:

- Trapezoidal fit between samples.
- Calculation of the square, and
- Calculation of the square root.

## 3. Methodology

The block diagram shown in Figure 2 depicts the steps to be followed in the computation of true RMS in the proposed method. The system consists of three functional blocks: Pre-process, Interactions and Post-process.

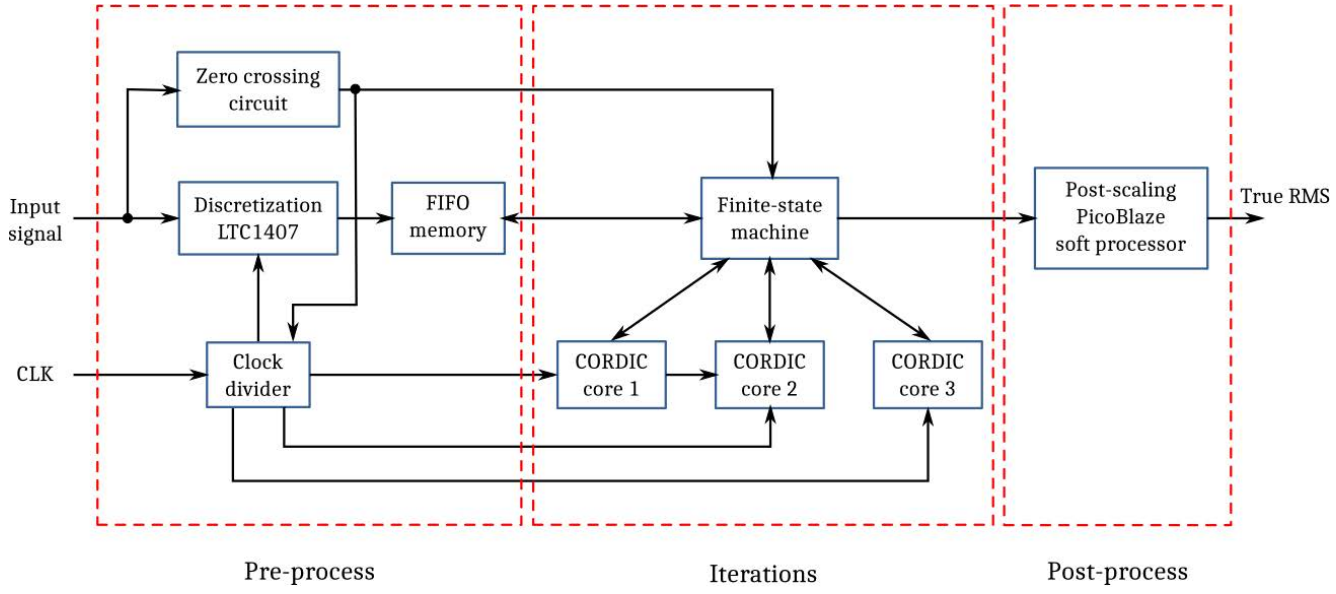


**Figure 2.** Hardware architecture.

The first block performs the conditioning of the input signal. There is a sub-block that is responsible for adjusting the voltage level using a resistive divider according to the specifications of the ADC (Analog-to-Digital Converter). This signal is then discretized to 14 bits with the ADC (Linear Technology LTC-1407-1A). This block also includes a zero crossing sensor circuit for calculating the period. This circuit is basically a voltage comparator synchronized with the clock signal.

The second block revolves around a Finite-State Machine (FSM) and the three CORDIC cores. In this block the trapezoidal estimation of the samples is performed, and the calculation of square and square root functions, one in each of the CORDIC cores.

In the case of the square root, we will think that the input value is given by equation 6 (the *a* is the input value). The *a* * *k*^2 is within the appropriate range for the CORDIC. We set *k* to 2 because the power of 2 can be realized by shift operation (ideal case for fixed point).

$$k\sqrt{a} = \sqrt{a * k^2} \tag{6}$$

Figure 3 shows a pseudo code of squared root operations in fixed point, and Figure 4 shows a performance comparison. Figure 4 functions are plotted for a relatively large sampling interval (20 us) to highlight their differences. The blue and red lines show the square root and the CORDIC square root, respectively. For short sampling intervals and/or larger intervals the two lines almost become one. Something similar is implemented for cases of trapezoidal interpolation and square.

```
 1  input a
 2  while a < s do
 3      a = a << 2
 4      m = m +1
 5  while a > t do
 6      a = a >> 2
 7      n = n + 1
 8  endwhile
 9  x = a + (1 << q)
10  y = a - (1 << q)
11  for i = 1 to j do
12      if y < 0 then
13          x = x + (y << i)
14          y = y + (x << i)
15      else y >= 0 do
16          x = x - y / (y << i)
17          y = y - x / (x << i)
18      endif
19  endfor
20      z = x * u
21      z << (n - m - r)
22  print z
```

**Figure 3.** CORDIC pseudo code in fixed point for square root.

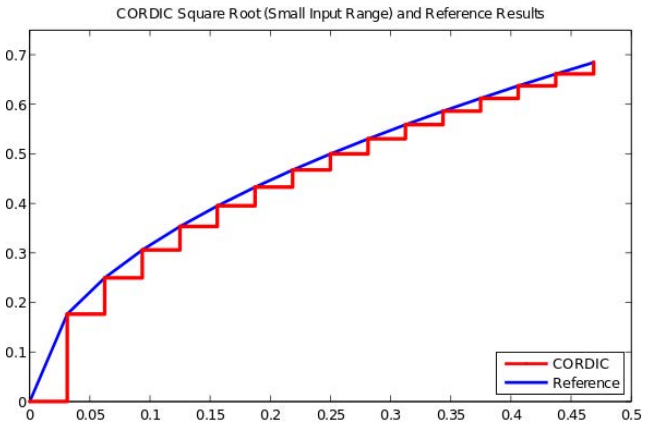


**Figure 4 .** Computer simulation vs CORDIC implementation result of the square root function.

The last block is formed by a PicoBlaze as processor embedded in the FPGA. This processor is responsible for the scaling of the output and its presentation on a LCD (Liquid-Crystal Display). The FPGA used in the prototype is the SPARTAN 3E 1600 from Xilinx.

## 4. Results and Discussion

The performance tests sought to determine the capability and versatility of the prototype. Therefore, we perform measurements on a set of known test signals, and we contrast the results against a calibrated laboratory equipment. The selected reference device is the Fluke 87, true RMS capable device. The reference signals are:

- Sine signal with frequencies from 1 to 20 kHz
- Square signal with frequencies from 1 to 20 kHz
- Triangular signal with frequencies from 1 to 20 kHz
- Constant DC signal of different values

Table 1 shows the results for 23 of these tests. The average error reached by the prototype against the references was 3.1%, while the Fluke 87 registered an average error of 5.8% with the same data. Future work will focus on applying a more rigorous testing protocol to the prototype.

The computational cost tests sought to determine the

**Table 1.** Comparison of performance against different test signals

| No_ data | Frequency [Hz] | Signal | Peak Voltage [V] | Prototype Reading [V] | Fluke 87 Reading [V] | Expected Value [V] | Error Prototype | Error Fluke 87 |
|---|---|---|---|---|---|---|---|---|
| 0 | 3061 | sine | 10 | 7,300 | 6,75 | 7,07 | 3,24% | 5% |
| 1 | 3061 | triangular | 10 | 5,976 | 5,76 | 5,77 | 3,51% | 0% |
| 2 | 3061 | square | 10 | 10,120 | 9,60 | 10,00 | 1,20% | 4% |
| 3 | 0 | dc | 10 | 10,230 | 9,80 | 10,00 | 2,30% | 2% |
| 4 | 60 | sine | 10 | 7,270 | 6,75 | 7,07 | 2,81% | 5% |
| 5 | 60 | triangular | 10 | 5,930 | 5,70 | 5,77 | 2,71% | 1% |
| 6 | 60 | square | 10 | 10,120 | 9,60 | 10,00 | 1,20% | 4% |
| 7 | 0 | dc | 10 | 10,223 | 9,80 | 10,00 | 2,23% | 2% |
| 8 | 60 | sine | 5 | 3,632 | 3,40 | 3,54 | 2,73% | 4% |
| 9 | 60 | triangular | 5 | 2,950 | 2,76 | 2,89 | 2,19% | 4% |
| 10 | 60 | square | 5 | 5,300 | 4,80 | 5,00 | 6,00% | 4% |
| 11 | 0 | dc | 5 | 5,200 | 4,98 | 5,00 | 4,00% | 0% |
| 12 | 4549 | sine | 5 | 3,290 | 3,10 | 3,54 | 6,94% | 12% |
| 13 | 4529 | sine | 10 | 7,310 | 6,68 | 7,07 | 3,38% | 6% |
| 14 | 4529 | triangular | 5 | 3,024 | 2,59 | 2,89 | 4,75% | 10% |
| 15 | 1000 | square | 5 | 5,017 | 4,55 | 5,00 | 0,34% | 9% |
| 16 | 4529 | square | 5 | 5,100 | 4,27 | 5,00 | 2,00% | 15% |
| 17 | 12930 | sine | 6,24 | 4,197 | 3,98 | 4,41 | 4,88% | 10% |
| 18 | 12930 | triangular | 6,24 | 3,390 | 3,28 | 3,60 | 5,90% | 9% |
| 19 | 12930 | square | 6,24 | 6,100 | 5,38 | 6,24 | 2,24% | 14% |
| 20 | 20000 | sine | 7,2 | 5,040 | 4,96 | 5,09 | 1,01% | 3% |
| 21 | 20000 | triangular | 7,2 | 4,178 | 4,09 | 4,16 | 0,51% | 2% |
| 22 | 20000 | square | 7,2 | 7,030 | 6,57 | 7,20 | 2,36% | 9% |

savings in prototype resources versus a traditional FPGA implementation without the use of CORDIC. In the initial tests we were able to determine a 31% reduction in No. of Slice LUTs (LookUp Tables), and 11% in No. of Slice Registers. In addition, the algorithm can be executed at a higher frequency.

## 5. Conclusion

The proposed methodology incorporates a true RMS algorithm and CORDIC algorithm. The inclusion of both techniques has reduced the computational complexity by of at least 30%, with a functional performance close to that identified in industrial equipment. The structure of the algorithm is simple and flexible, allowing its adaptation in a large number of applications of energy quality processing, in particular in power factor active correctors.

## 6. Acknowledgment

This work was supported by the District University Francisco José de Caldas, in part through CIDC, and partly by the Technological Faculty. The views expressed in this paper are not necessarily endorsed by District University. The authors thank the research groups ARMOS and DIGITI for the evaluation carried out on prototypes of ideas and strategies.

## 7. References

1. Pramanik S, Chakraborty S, Saha R, Basu R, De R, Chatterjee S. Low Latency High Throughput CORDIC based Fourier Analysis. IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON 2016), 2016. p. 1–3. Crossref
2. Xiaoning L, Yizhuang X, He C, Bingyi L. Implementation on FPGA for CORDIC-based Computation of Arcsine and Arccosine. IET International Radar Conference, 2015. p. 1–4. Crossref
3. Bertel O, Moreno C, Toro E. Aplicación de la transformada Wavelet para el reconocimiento de formas en visión artificial. Tekhnê. 2009; 6(1):3–8.
4. Andraka R. A survey of CORDIC algorithms for FPGA based computers. Proceedings of the 1998 ACM/SIGDA Sixth International Symposium on Field Programmable Gate Arrays, 1998. p. 191–200. Crossref
5. Subhashis R, Joyashree B, Sarkar S. Design and VLSI Implementation of a Robot Navigation Processor deploying CORDIC based Anti-collission Algorithm with RFID Technology. Annual IEEE India Conference, 2014 Dec. p. 1–6.
6. Rih-Lung C, Yi-Qin Z, Shih-Lun C. Fully pipelined CORDIC-based inverse kinematic FPGA design for biped robots. Electronics Letters. 2015 Aug; 51(16):1241–3. Crossref
7. Muchahary D, Mondal A, Majumder A. A CORDIC Based Design Technique for Efficient Computation of DCT. International Conference on Communications and Signal Processing, 2015 Apr. p. 1–6. Crossref
8. Kusumah R, Cahyadi D, Kumara G, Adiono T. CORDIC-Based Digital Sound Synthesizer. IEEE Region 10 Conference, 2015 Nov. p. 1–6.
9. Vyas P, Vachhani L, Sridharan K, Pudi V. CORDIC-Based Azimuth Calculation and Obstacle Tracing via Optimal Sensor Placement on a Mobile Robot. IEEE/ASME Transactions on Mechatronics. 2016; 21(5):2317–29. Crossref
10. Huijie Z, Yizhou G. CORDIC-based implementation architectures for the second-harmonic excitation of a micro-gyroscope. IEEE 11th Conference on Industrial Electronics and Applications, 2016 Oct. p. 2107–10.
11. Heidarpour M, Ahmadi A, Rashidzadeh R. A CORDIC Based Digital Hardware For Adaptive Exponential Integrate and Fire Neuron. IEEE Transactions on Circuits and Systems I. Regular Papers. 2016 Sep; 63(11):1986–96. Crossref
12. Ramadoss R, Mozaffari M, Azarderakhsh R. Reliable Hardware Architectures of CORDIC Algorithm with Fixed-Angle of Rotations. IEEE Transactions on Circuits and Systems II, Express Briefs. 2016 Nov; (99):1. Crossref
13. Kumar V, Chandra K, Kumar P. Low-complexity CORDIC-based VLSI Design and FPGA Prototype of CI-OFDMA System for Next-generation. IEEE 12th International Colloquium on Signal Processing & Its Applications, 2016 Jul. p. 22–7.
14. Tang A, Yu L, Han F, Zhang Z. CORDIC-based FFT Real-time Processing Design and FPGA Implementation. IEEE 12th International Colloquium on Signal Processing & Its Applications. 2016 Jul. p. 233–6.
15. Meher P, Valls J, Tso-Bing J, Sridharan K, Maharatna K. 50 Years of CORDIC: Algorithms, Architectures, and Applications. IEEE Transactions on Circuits and Systems I: Regular Papers. 2009 Jun; 56(9):1893–907. Crossref