FPGA Based Design of Digital Wave Generator Using CORDIC Algorithm

Rajesh Mehra[1], Bindiya Kamboj[2]
[1]Faculty, Department of Electronics & Communication Engineering
[2]ME Student, Department of Electronics & Communication Engineering
National Institute of Technical Teachers' Training & Research, Sector-26, Chandigarh, India
[1]rajeshmehra@yahoo.com, [2]bindiyakamboj@gmail.com

Abstract

In this paper an efficient approach is presented to design and implement a high speed and area efficient Digital Sine and Cosine Wave Generator for wireless applications like SDR and GSM. The implementation is based on **C**oordinate **R**otation **D**igital **C**omputer (CORDIC) algorithm which uses shifts, additions and a very small lookup table (LUT). It is an efficient method used to compute trigonometric functions, multiplications, divisions, data type conversions and hyperbolic functions in a simple and elegant way. The proposed design is synthesized with ISE 10.1i software, simulated with Modelsim 6.3XE simulator and implemented on Spartan-3 FPGA target device using Chipscope 10.1. Results show enhanced performance using proposed CORDIC algorithm in terms of speed, area and power utilization.

Key Words: FPGA, CORDIC, Simulation, VHDL, Pipeline

1. Introduction

Digital signal processing (DSP) algorithms exhibit an increasing need for the efficient implementation of complex arithmetic operations. The computation of trigonometric functions, coordinate transformations or rotations of complex valued phasors is almost naturally involved with modern DSP algorithms. Popular application examples are algorithms used in digital communication technology and in adaptive signal processing. While in digital communications, the straightforward evaluation of the cited functions is important, numerous matrix based adaptive signal processing algorithms require the solution of systems of linear equations, QR factorization or the computation of eigen values, eigenvectors or singular values. All these tasks can be efficiently implemented using processing elements performing vector rotations. The CORDIC algorithm is used to compute the trigonometric functions, multiplications, divisions, data type conversions, and hyperbolic functions. Two basic CORDIC modes are known leading to the computation of different functions, the rotation mode and the vectoring mode. For both modes the algorithm can be realized as an iterative sequence of additions/subtractions and shift operations, which are rotations by a fixed rotation angle but with variable rotation direction. Due to the simplicity of the operations involved, the CORDIC algorithm is well suited for VLSI implementation.

In this paper CORDIC algorithm is used to design a digital sine and cosine waveform generator. There are plenty of applications which require digital wave generators. Wireless and mobile systems are among the fastest growing application areas; in particular, Software Defined Radio (SDR) is currently a focus of research and development. An SDR system allows performing many functions based on a single hardware platform, thus highly reconfigurable resources for signal processing are needed, mainly for modulation and demodulation of digital signals [1]. Fourth generation (4G) wireless and mobile systems are currently the focus of research and development. They will allow new types of services to be universally available to consumers and for industrial applications. Broadband wireless networks will enable packet based high data rate communication suitable for video transmission and mobile Internet applications [2].

2. CORDIC Algorithm

In the most general form, one CORDIC iteration can be written as given below

$$X_{i+1} = x_i - m.\mu_i.y_i.\delta_{m,i}$$
$$Y_{i+1} = y_i - m.\mu_i.x_i.\delta_{m,i} \quad (1)$$
$$Z_{i+1} = z_i - m.\mu_i.\alpha_{m,i}$$

In order to avoid multiplication $\delta_{m,I}$ is defined as

$$\delta_{m,I} = d^{-s_{m,i}}$$
$$\delta_{m,I} = 2^{-s_{m,i}} \quad (2)$$

where

d = Radix of employed number system

$s_{m,i}$ = integer number

$2^{-s_{m,i}}$ = Radix 2 number system

The first two equations of Eq. (1) in a matrix-vector product form

$$v_i = \begin{pmatrix} 1 & -m.\mu_i.\delta_{m,i} \\ m.\mu_i.\delta_{m,i} & 1 \end{pmatrix}.v_i \quad (3)$$

Hence

$$v_{i+1} = C_{m,i}.v_i \quad (4)$$

To verify the matrix-vector product in Eq. (3), let us consider a general normalized plane rotation matrix for the three coordinate systems

$$R_{m,i} = \begin{pmatrix} \cos(\sqrt{m}.\alpha_{m,i}) & -\mu i.\sin(\sqrt{m}.\alpha_{m,i}) \\ \frac{\mu}{\sqrt{m}}.\sin(\sqrt{m}.\alpha_{m,i}) & \cos(\sqrt{m}.\alpha_{m,i}) \end{pmatrix} \quad (5)$$

For m = 1,0,-1 and an angle $\mu_i.\delta_{m,i}$ with μ_i determining the rotation direction and $\delta_{m,i}$ representing an unsigned angle. Figure1 shows the rotation trajectory for the circular coordinate system with m = 1.

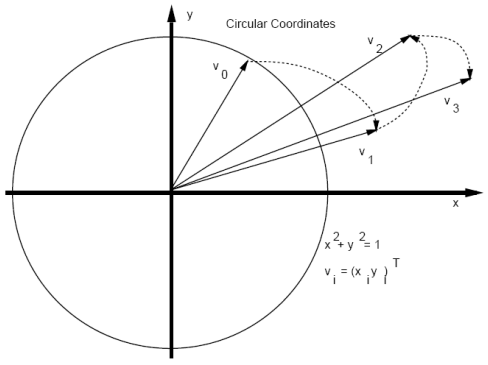

Figure1. Rotation Trajectory

Dividing both sides of Eq(5) by $\cos(\sqrt{m}.\alpha_{m,i}$ then we get

$$\frac{1}{\cos(\sqrt{m}.\alpha_{m,i})}.Rm,i = \begin{pmatrix} 1 & -\mu_i.\sqrt{m}.\tan(\sqrt{m}.\alpha_{m,i}) \\ \frac{\mu_i}{\sqrt{m}}.\tan(\sqrt{m}.\alpha_{m,i}) & 1 \end{pmatrix}$$

Hence

$$\frac{1}{\cos(\sqrt{m}.\alpha_{m,i})}.Rm,i = C_{m,i}$$

for

$$\delta_{m,i} = \frac{1}{\sqrt{m}}.\tan(\sqrt{m}.\alpha_{m,i})$$

This proves that $C_{m,i}$ is an un-normalized rotation matrix for m ε {-1,1}due to the scaling factor. The scaling factor is given by

$$K_{m,i} = \frac{1}{\cos(\sqrt{m}.\alpha_{m,i}}$$
$$= \frac{\sqrt{\cos^2(\sqrt{m}.\alpha_{m,i}) + \sin^2(\sqrt{m}.\alpha_{m,i})}}{\cos(\sqrt{m}.\alpha_{m,i}} \quad (6)$$
$$= \sqrt{1+\tan^2(\sqrt{m}.\alpha_{m,i})}$$

For n successive iterations

$$v_n = \prod_{i=0}^{n-1} C_{m,i}.v_0 = \prod_{i=0}^{n-1} K_{m,i}.\prod_{i=0}^{n-1} R_{m,i}.v_0 \quad (7)$$

i.e. a rotation by an angle θ is given by

$$\theta = \sum_{i=0}^{n-1} \mu_i.\alpha_{m,i} \quad (8)$$

which is performed with an overall scaling factor of

$$K_m(n) = \prod_{i=0}^{n-1} K_{m,i} \quad (9)$$

The third iteration component z_i simply keeps track of the overall rotation angle accumulated during successive micro rotations.

$$Z_{i+1} = z_i - m.\mu_i.\alpha_{m,i} \quad (10)$$

After n iterations

$$z_n = z_0 - \sum_{i=0}^{n-1} \mu_i.\alpha_{m,i} \quad (11)$$

Hence z_n is equal to the difference of the start value z_0 and the total accumulated rotation angle.

3. Circuit Description

Figure2 shows the CORDIC pipelined architecture for one iteration. All internal variables are represented by a fixed number of digits, including the pre calculated angle $\alpha_{m,i}$ which is taken from a register. Due to the limited word length some rounding or truncation following the shifts $2^{-Sm,i}$ is necessary [3]. The adders/subtractors are steered with $-m\mu_i$, μ_i and $-\mu_i$, respectively.

A rotation by any desired angle A_0 can be achieved by defining a converging sequence of n single rotations. The CORDIC algorithm is formulated by

1. A shift sequence $s_{m,i}$ defining an angle sequence

$$\alpha_{m,i} = \frac{1}{\sqrt{m}}.\tan^{-1}(\sqrt{m}.2^{-Sm,i}) \qquad (12)$$

with i ε {0,1…n-1} gives convergence.

2. A control scheme generating a sign sequence μ_i with i Є{0,1…n-1} which steers the direction of the rotations in this iteration sequence and guarantees convergence. The angle Ai is introduced specifying the remaining rotation angle after rotation i, is given by

$$|A_{i+1}| = ||Ai| - \alpha_{m,i}| \qquad (13)$$

Two control schemes fulfilling Eq. (13) are known for the CORDIC algorithm, the rotation mode and the vectoring mode.

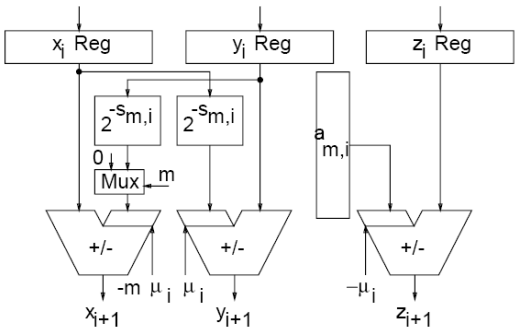

Figure2. Pipelined CORDIC Architecture

In order to reduce the number of clock cycles pipeline registers are used in the architecture.

4. Proposed Design Simulation

In this proposed work, rotation mode is used to design the digital sine and cosine waveform generator. In rotation mode the desired rotation angle $A_0 = \theta$ is given for an input vector $(x; y)^T$. We set $x_0 = x$, $y_0 = y$, and $z_0 = \theta = A_0$. After n iterations

$$z_n = \theta - \sum_{i=0}^{n-1} \mu_i.\alpha_{m,i} \qquad (14)$$

In figure 3, the trajectory for the rotation mode in the circular coordinate system is shown. It becomes clear that the vector is iteratively rotated towards the desired final position. The scaling involved with the successive iterations is also shown. Here we have chosen angles that have tan powers of 2.

We have used series of angles to form all other angles i.e. 45, 26.6, 14, 7.1, 3.6, 1.8, .9, 0.4. For example

a) 30 = 45 -26.6+14 -7.1+3.6+1.8 -0.9+0.4
b) 90 = 45+26.6+14+7.1-3.6+1.8 -0.9+0.4

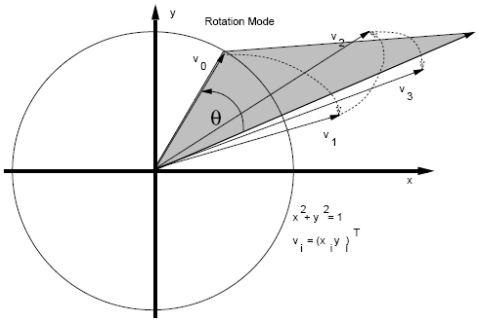

Figure3. Rotation Mode Trajectory

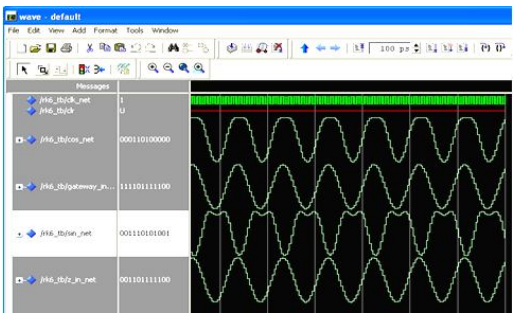

Figure4. Model sim Based Simulation

The proposed design was synthesized on Spartan3 based xc3s200-5 ft 256 FPGA device using Xilinx ISE 10.1 and simulated using modelsim [4] and hardware co-simulated chipscope [5,6]. The Modelsim and Chipscope based simulated outputs have been shown in Figure4 and Figure5 respectively.

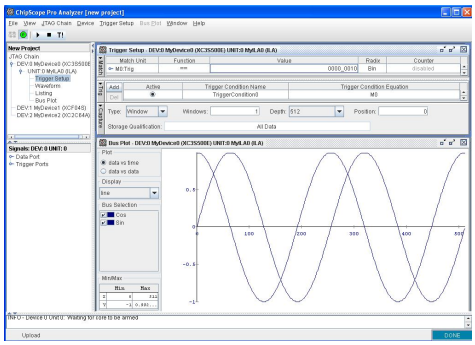

Figure5. Chipscope Based H/W Co-Simulation

Figure6 shows the output verification of chipscope results with the help of Matlab [7]. It can be observed both results are matching with each other.

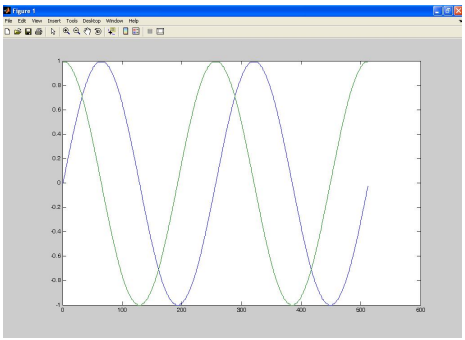

Figure6. Matlab Based O/P Verification

5. Hardware Synthesis Results

Table1 shows the area utilization of proposed design implemented on xc3s200-5ft256 target device in terms of slices, flip-flops, LUTs, IOBs and GCLK.

Table1. Area Utilization

Device Utilization Summary (estimated values)			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	196	1920	10%
Number of Slice Flip Flops	321	3840	8%
Number of 4 input LUTs	361	3840	9%
Number of bonded IOBs	37	173	21%
Number of GCLKs	1	8	12%

It can be seen from Figure7 that the proposed design uses very less number of available resources of target device.

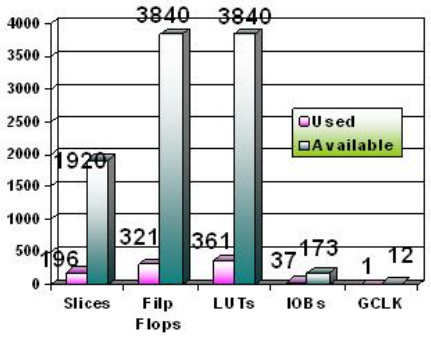

Figure7. Resource Utilization

The total power consumption of the proposed design is 0.03942 W at 26.2 degrees C as shown in Table2.

Table2. Power Consumption

Name	Value	Used	Total Available
IOs	0.00022 (W)	37	173
Total Quiescent Power	0.03722 (W)		
Total Dynamic Power	0.00219 (W)		
Total Power	0.03942 (W)		
Junction Temp	26.2 (degrees C)		

The proposed design operates at maximum frequency of 161.65 MHz whereas the design shown in [1] operates at a maximum frequency of 154.69 MHz. So the proposed design shows an improvement in speed with considerable reduction in used resources on target device.

6. Conclusion

In this paper, a pipelined CORDIC algorithm based design of digital sine and cosine wave generator for modulation and demodulation in wireless communication is presented. As compared to other techniques, CORDIC algorithm based designs show better results in terms of speed and area utilization when applied to wireless applications like SDR and GSM. The proposed design operates at a maximum frequency of 161.65 MHz along with efficient power and area utilization to provide cost effective solution for wireless applications.

Acknowledgements

The author would like to thank Dr. Swapna Devi, Associate Professor, Electronics & Communication Department, NITTTR, Chandigarh for constant encouragement, and guidance during this research work. The author would also like to express his sincere thank and deep sense of gratitude to Dr. S. Chatterji, Professor and Head, Electronics & Communication Department and Dr. S.S.Pattnaik, Professor & Head, ETV Department, NITTTR, Chandigarh for their constant inspirations, support and helpful suggestions throughout this research work**.**

References

[1] Esteban O. Garcia, Rene Cumplido, Miguel Arias" Pipelined CORDIC Design on FPGA for a Digital Sine and Cosine Waves Generato," International conference on Electrical and Electronics Engineering (ICEEE), pp 1-4, IEEE 2006.

[2] Eckhard Grass, Bodhisatya Sarker and Koushik Maharatna "A Dual-Mode Synchronous / Asynchronous CORDIC Processor," IEEE International Symposium on Asynchronous Circuits and Systems, pp. 76-86 , IEEE 2002.

[3] Javier Valls, Martin Kuhlmann, and Keshar K. Parhi. "Evaluation of CORDIC algorithms for FPGA design," *Journal of VLSI Signal Processing*. vol. 32, no. 3, pp. 207-222, 2002.

[4] ChipScope Pro 10.1 Serial I/O Toolkit User Guide UG213 (v10.1) March, 2008.

[5] ChipScope Pro 10.1 Software and Cores User Guide UG029 (v10.1) March, 2008.

[6] ModelSim® User's Manual Software Version 6.2g February 2007.

[7] Communications Toolbox™ User's Guide, March 2007.