Complex DPLL Using CORDIC Processor for I-Q Channel Demodulator

Prof.R.V.Babar[1]
Asst.Prof. Dept. of Electronics and Telecommunication
Sinhgad Institute of Technology(SIT)
Lonavala,Pune.maharashtra india
rajendra_babar@yahoo.com

Dr.M.S.Gaikwad[2]
Principal,Sinhgad Institute of Technology(SIT)
Lonavala,Pune.maharashtra india
drmsgaikwad@gmail.com

Ms.Urmila More[3]
Dept. of Electronics and Telecommunication
(VLSI and Embedded systems)
Sinhgad Institute of Technology(SIT)
Lonavala,Pune. .maharashtra india
urmi_more@rediffmail.com

***Abstract*—The aim of this paper is to present FPGA (Field Programmable Gate Array) based implementation of CORDIC (Co-ordinate Rotation Digital Computer) processor for complex DPLL(Digital Phase Locked Loop) for use in I-Q channel demodulator includes phase detector for producing a phase error indicative of a difference in phase between a complex digital input signal and complex digital feedback signal .This structure is very suitable for VLSI implementations. In the first part of the paper, we describe operation of the design of pipelined architecture for coordinate rotation algorithm. The design of CORDIC in the vector rotation mode results in high system throughput due to its pipelined architecture. Then a CORDIC realization of a complex phase-locked loop is introduced. Finally we discuss the effects of the number of iterations to the performance of this kind of phase-locked loop.**

Keywords - CORDIC Pipelined Architecture, DPLL, Micro-rotation.

I. INTRODUCTION

CORDIC algorithm was first developed by Jack E. Volder in 1959 [1]. CORDIC algorithm is xtremely useful in efficient and effective implementation of DSP systems [2]. This algorithm allows implementation of trigonometric functions like sine, cosine, magnitude and phase with great precision by using just simple shift and adding operations [1-4]. Although the same functions can be implemented using multipliers, variable shift registers or Multiply Accumulator (MAC) units, but CORDIC can implement these functions efficiently while saving enough silicon area which is considered to be a primary design criteria in VLSI technology.

This paper designs first order complex DPLL using CORDIC which functions as a FSK demodulator. In digital PLL, an adjustable local sine wave generator and phase detector is required. The sine and cosine terms can be calculated using polynomial approximation, e.g. Taylor series.But it requires a considerable amount of hardware space on the silicon substrate. Interpolation method using table look-up may be the other solution. But it also requires large number of gates and ROM memory. The CORDIC offers the opportunity to calculate the desired functions in a simple and efficient way.Due to the simplicity of the involved operations, the CORDIC realization of complex DPLL is very well suited in VLSI hardware design and its implementation. This paper first describes the CORDIC algorithm and then pipelined architecture design [9]. Thereafter, implementation with adjustment of micro-rotation has been described. Finally CORDIC realization of a complex phase locked loop is described.

II. CORDIC ALGORITHM

There exist two modalities of CORDIC algorithm, vectoring and rotation mode. In vectoring mode, coordinates (Xo,Yo) are rotated until Yo converges to zero. In rotation mode, initial vector (Xo,Yo) starts aligned with the x axis and is rotated by an angle of Θ_i every cycle, so after n iterations,Θ_n is the obtained angle. For this work, rotation mode is used to approximate sine and cosine functions.The way CORDIC works is related to trigonometric function properties. The main idea consists in taking a unary vector and applying successive rotations, called micro rotations, until the desired angle is reached. The rotating vector is unary, so after n iterations it will contain in Θ_n and cos Θ_n in its second and first components respectively. The calculus can be simplified if the starting vector is approximated to a constant value K. Starting vector is defined as Va = (Xo,Yo). In order to make the vector rotation, a linear transform, which can be described by a matrix which is multiplied by a vector, is used. After n iterations, the vector (x_n,y_n) is

$$Xn = Xo \cos \Theta - Yo \sin \Theta \quad (1)$$
$$Yn = Xo \cos \Theta + Xo \sin \Theta \quad (2)$$

In each iteration i, the vector perform a micro rotation by Θ_i ,so the new vector is calculated with a similar function:

$$X_{i+1} = X_i \cos \Theta_{i+1} - Y_i \sin \Theta_{i+1} \quad (3)$$
$$Y_{i+1} = Xo \cos \Theta_{i+1} + Xo \sin \Theta_{i+1} \quad (4)$$

When the term $\cos \Theta_{i+1}$ is factorized, components in the vector are described by:

$$X_{i+1} = \cos \Theta_{i+1} (X_i - Y_i \tan \Theta_{i+1}) \quad (5)$$
$$Y_{i+1} = \cos \Theta_{i+1} (X_i + Y_i \tan \Theta_{i+1}) \quad (6)$$

tan Θ_{i+1} is restricted to $\pm 2^{-i}$, so multiplication is converted in an arithmetic right shift. It is also useful to use the identity

$$\cos \Theta_{i+1} = \cos(\arctan 2^{-i})$$

to define the variables as:

$$K_i = \cos(\arctan 2^{-i}) = 1/\sqrt{(1+2^{-2i})} \quad (7)$$
$$d_i = \pm 1 \quad (8)$$

Cosine is an even function, therefore

$$\cos(\alpha) = \cos(-\alpha)$$

So (5) and (6) can be transformed into:

$$X_{i+1} = K_i(X_i - Y_i d_i 2^{-i}) \quad (9)$$
$$Y_{i+1} = K_i(X_i + Y_i d_i 2^{-i}) \quad (10)$$

Multiplication by K_i is avoided by considering it as a gain factor for all iterations. If n iterations are performed, then K is defined as the multiplication of every K_i .

$$K = \pi K_i = \pi\, 1/\sqrt{(1+2^{-2i})} \quad (11)$$

As the vector is initialized with constant K, the vector components for each iteration are simplified to:

$$X_{i+1} = X_i - Y_i d_i 2^{-i} \quad (12)$$
$$Y_{i+1} = Y_i + X_i d_i 2^{-i} \quad (13)$$

On each iteration it is necessary to decide whether $d_i = 1$ or $d_i = -1$. In order to make that decision, the difference between the desired angle and the current angle is used. So a new variable known as accumulator is defined as

$$Z_{i+1} = Z_i - d_i \arctan 2^{-i} \quad (14)$$

The value of Zo is the angle for which sine and cosine is to be calculated. To know whether d; should be positive or negative, the following rule is used:

$$d_i = \begin{cases} -1 & \text{Sin} Z_i < 0 \\ +1 & \text{Sin} Z_i \geq 0 \end{cases} \quad (15)$$

The sum of the rotating angles gives the desired angle

$$\Theta_n = \sum d_i \arctan 2^{-I} \quad (16)$$

Because the first tangent value is 2°= 1 it is possible to ,rotate angles only in the range $[-\pi/2, \pi/2]$.

III. PIPELINE CORDIC ARCHITECTURE

Modules are incorporated and each module is responsible for one elementary rotation. The modules are cascaded through intermediate latches Every stage of pipe lined CORDIC architecture only adder/subtract is used. The shift operations are hardwired using, permanent oblique bus connections, to perform multiplications by $2^{-i.}$, The pre computed values, as given in Table I, of i-th iteration angle α_i, required at each module, is stored at a ROM memory location. The delay is adjusted by using proper bit-length in the shift register. Since no sign detection is needed to force $Z_f = 0$, the carry save adders are well suited in this architecture. The use of this compressed arithmetic reduces the stage delay significantly. With the pipelining architecture, the propagation delay of the multiplier is the total delay of a single adder. Therefore, the throughput of the architecture is increased to many folds as the throughput is given by:

$$\frac{1}{\text{Delay due to a single adder}} \quad (17)$$

If an iterative implementation of the CORDIC is used, the processor would take several clock cycles to give output for a given input. But in the pipelined architecture, each pipeline stage takes exactly one clock cycle to pass one output.

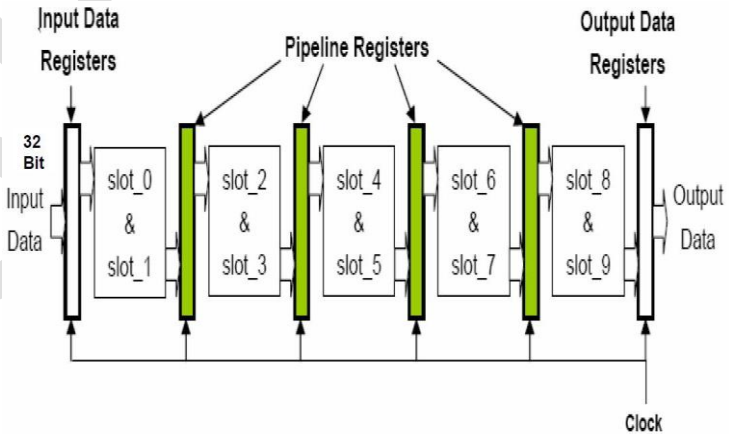
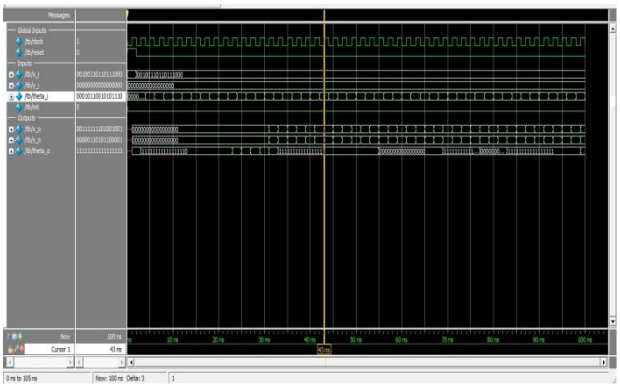
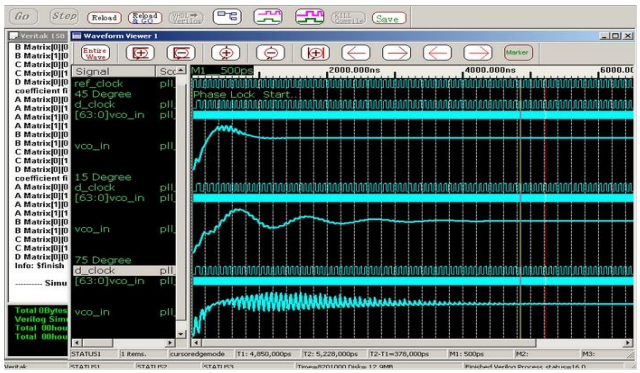


Fig.1 Pipelined CORDIC Architecture

IV. SIMULATION RESULT



CORDIC TOP LEVEL OUTPUT



DPLL TOP LEVEL OUTPUT

V. COMPLEX DIGITAL PHASE LOCKED LOOP USING CORDIC

The down converted and filtered baseband signal has two components: real and imaginary parts. Therefore, the base band signal can hold both the amplitude and phase of the sinusoidal signal at the same time. It does not hold any image frequency. So only lower order loop filter is sufficient for the DPLL. This reduces the complexity of already complex DPLL design. The main advantage is that no group delay is caused by the loop filter. Using the vector rotation operator [x, y]T ∠Θ, the complex first-order DPLL demodulator equations for a given input signal can be stated as:

a). The real part of the output in phase comparator equation : $\varepsilon_n = R\{s(n) \angle - \Theta_n\}$

b).The loop filter equation $C_n = 2\pi k \varepsilon_n$ where K is the loop filter coefficient.

The loop filter coefficient K depends on the sampling frequency and number of iterations of CORDIC algorithm. For M number of iterations, the loop filter coefficient K can be given by:

$$K_{cordic} = \frac{K}{\prod_{i=0}^{M-1} (1 + 2^{-2i})^{1/2}}$$

c) The phase accumulator equation:

$$\Theta_{n+1} = (\Theta_n + 2\pi k \varepsilon_n + \omega c) \bmod 2\pi$$

Where $\omega c = 2\pi f_c$ is the center frequency.

The CORDIC based DPLL tries to adjust the continuous phase rotation in such a way that the complex component of the rotated vector will always be zero. So to get the required loop performance, we can set the input signal of the CORDIC as: $R\{s(n)\} = x_n$ and $I\{s(n)\} = y_n$ The algorithm, architecture and convergence of the CORDIC has been already discussed. The number of iterations in the CORDIC algorithm determines the speed and accuracy of the CORDIC based Digital PLL. We have designed a pipelined CORDIC architecture which is able to provide better loop performance and increased SNR at numbers of iterations.

Table I. Pre-Computed Angles

I	$2^{-i} = \tan\alpha_i$	$\alpha_i = \arctan(2^{-i})$	α_i in radian
0	1	45°	0.7854
1	0.5	26.565°	0.4636
2	0.25	14.063°	0.2450
3	0.125	7.125°	0.1244
4	0.0625	3.576°	0.0624
5	0.03125	1.7876°	0.0312
6	0.015625	0.8938°	0.0156
7	0.0078125	0.4469°	0.0078
..	.....	.....	.....

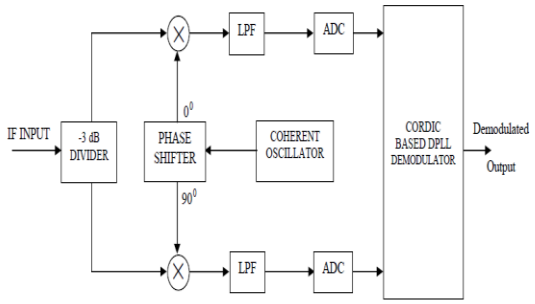


Fig.2- I/Q channel Processor based complex DPLL Demodulator

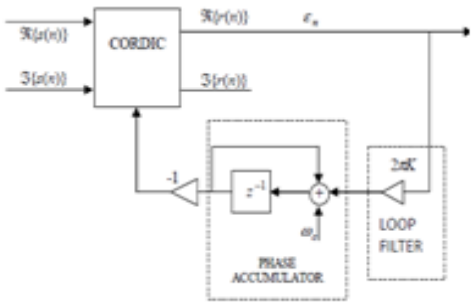


Fig.3- Complex DPLL using CORDIC

VI. I/Q CHANNEL DEMODULATION

As per the Runge Kutta theorem, vector sum of cosine's component is completely real whereas the spectrum of sine component is totally imaginary. If the cosine and sine components are combined, the resultant spectrum becomes one sided with direction of rotation (positive or negative frequency) and with known real (cosine) and imaginary (sine) components [7].

$$\cos\omega_c t + j \sin\omega_c t = e^{j\omega ct}$$

The process of recovering both real and imaginary signal component is known as *I/Q* demodulation. *I* stand for in-phase
channel which processes cosine (real) components. *Q* stands for in-quadrature channel which processes sine (imaginary) component. The input of *I/Q* channel is Intermediate Frequency (IF). If the carrier frequency of IF is *fc* with a time
varying amplitude $a(t)$ and time varying phase $\varphi(t)$,
then input signal $s(t)$ will be :

$$S(t) = a(t) \cos[(2\pi fc\, t + \varphi(t)] .$$

In *I* channel, the IF signal is multiplied by reference carrier frequency produced by crystal oscillator at zero phase reference. The output of the *I* channel mixer is, *I(t)*
given by :

$$I(t) = a(t) \cos[(2\pi fc\, t + \varphi(t)].\cos(2\pi fc\, t)$$
$$= a(t) \cos[\varphi(t)] + a(t) \cos[4\pi f\, c\, t + \varphi(t)]$$

The first term is the average value (DC) of the product and represents cosine of the signal phase and amplitude. The second term with high frequency component is suppressed by
Low Pass Filter (LPF). So the output of the *I* channel is

$$I(t) = a(t) \cos[\varphi(t)] .$$

Similarly, the *Q* channel output can be derived. The LPF output at *Q* channel is:

$$Q(t) = a(t) \sin[\varphi(t)] .$$

Thus, *I* and *Q* channel together provide the amplitude and phase modulation.

CONCLUSION

The use of pipelined CORDIC computational architecture makes implementation of this kind of demodulator easier. Numbers of micro-rotations have been adjusted so as to achieve better loop performance and speed of operation while minimizing angle approximation error. CORDIC algorithm is used to achieve high throughput facilitating real time signal processing, by using complex Digital Phase Locked Loop.

REFERENCES

[1] J.E. Volder. "The CORDIC Trigonometric Computing Technique". *IRE Transactions on Electronic Computing*, vol EC-8, pp 330-334, Sept 1959.
[2] Y.H. Hu. "CORDIC-Based VLSI Architectures for Digital Signal Processing" *IEEE Signal Processing Magazine*, Vol. 9, No. 3, pp. 16- 35, 1992.
[3] Andraka R.A., "Survey of CORDIC Algorithms for FPGA Based Computers", *Proceedings of the 1998 ACM/SIGDA 6th International Symposium on FPGAs*, pp 191-200, Monterey, California, Feb.22-24, 1998.
[4] S. Wang, V. Piuri, E. E. Swartzlander. Jr., "Granularly-pipelined CORDIC processors for sine and cosine generators", IEEE International Conference on *Acoustics, Speech, and Signal Processing ICASSP*, vol. 6, pp. 3298-3301, 1996.
[5] Y.H. Hu. "The Quantization Effects of the CORDIC Algorithm"*IEEE Trans. Signal Processing*, vol. 40, No. 4, pp. 834-844, Apr. 1992.
[6] M. Chakraborty, A. S. Dhar and Moon Ho Lee, "A Trigonometric Formulation of the LMS algorithm for Realisation of Pipelined CORDIC," *IEEE Trans. Circuits and Systems*, vol. 52, no. 9, pp. 530-534, Sep.2005.
[7] Byron Edde, "Radar Principles, Technology, Applications" *Pearson Education,Inc*. ISBN 978-81-317-1383-9.
[8] Vuori J., "Implementation of a Digital Phase-Locked Loop Using CORDIC Algorithm", *IEEE International Symposium on Circuit and Systems*, Atlanta, USA 1996, pp. IV-164-167.
[9] A. Mandal, K.C. Tyagi, B.K. Kaushik, "VLSI Architecture Design and Implementation for Application Specific CORDIC Processor", *2nd IEEE International Conference on Advances in Recent Technologies in Communicaion and Computing (ARTCom)*,pp 191- 193, Oct 16-17,2010