FPGA Implementation of Pipelined CORDIC for Digital Demodulation in FMCW Radar

Amritakar Mandal, Rajesh Mishra

Abstract— **Now-a-days Radar Signal Processing system is gaining a great deal of attention for realization of on-chip programmable signal processor for its real time applications. Application specific systems are being implemented using wide spectrum of Digital Signal Processing (DSP) algorithms. Such is the case for COordinate Rotation DIgital Computer (CORDIC) algorithm which is turned out to be widely researched topic in the field of vector rotated DSP applications. In this paper we have designed an application specific pipelined CORDIC architecture for digital demodulation in low power, high performance FMCW Radar. A complex Digital Phase Locked Loop (DPLL) has been used for digital demodulation. The FPGA implementation of CORDIC based design is suitable because of its inherent high system throughput due to its pipelined architecture where latency is reduced in each of the pipelined stage. Substantial amount of resource utilization has been reduced in proposed design. For better loop performance of first order complex DPLL during demodulation, the convergence of the CORDIC architecture is also optimized. Hardware synthesized result using Cadence design tools are presented.**

Index Terms— **FMCW Radar, CORDIC, FPGA, DSP, DPLL, Loop performance.**

I. Introduction

PHASE DETECTION is a widely researched topic in radar demodulation, especially in low power Doppler radar where accurate detection is needed [1]. In general, radar system uses coherent oscillator as a reference frequency for Doppler detection. In analog demodulation, Voltage Control oscillator (VCO) is used. But perfect demodulation is not possible with the use of VCO as it suffers to maintain linearity over the desired frequency range [2]. In present decade, digital or mixed signal design based demodulator is widely used for superior performance. To ensure linearity over the range of frequency, numerically controlled oscillator is being used. Phase detection in communication receiver is very much sensitive to quantization noise. This kind of distortion is basically due to bit resolution. Efforts have been made to design a quantization error free pipelined CORDIC architecture based digital demodulator on FPGA platform. We have proposed a first order complex Digital Phase Locked Loop (DPLL) for efficient demodulation of radar signals.

The iterative formulation of CORDIC algorithm was first developed by Jack E. Volder in 1959 [3] for the multiplication, division and computation of trigonometric functions like sine, cosine, magnitude and phase with great precision. The key concept of CORDIC algorithm is simply shift and adds. 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 application specific on chip implementation where high performance and low cost hardware solutions for DSP are required [4]. This kind of demodulator can be used in wide variety of receivers where real time signal processing is important [5].

This paper designs first order complex DPLL for I/Q channel Radar demodulator using pipelined CORDIC architecture. In digital PLL, an adjustable local sine wave generator and phase detector is required. The CORDIC offers the opportunity to calculate the desired trigonometric computation in a simple and efficient way. Due to the simplicity of the involved operations, the CORDIC realization of complex DPLL is very well suited for on-chip hardware design and its implementation. The analysis of various error sources is necessary for optimal design of system using the CORDIC processor. In DSP systems, signals are required to be quantized and represented in fixed word-length. A limited word-length results in the round-off noise and degradation of Signal-to-Quantization Noise Ratio (SQNR) performance [6,7]. In general, larger the dynamic range of the signals, more severe is the round-off noise. To reduce the computation error, a processor designer might simply increase the number of iterations and that will be a huge wastage of processing time and power. Therefore, exact computation of word-length is necessary for designing an architecture for CORDIC. If word-length is larger, the computational speed of CORDIC reduces significantly [8,9]. On the other hand, if we implement with smaller word-length, the design will suffer from danger of overflow. To design an optimal application specific CORDIC processor for a high performance signal processing, the choice of word-length and number of iterations in error analysis are needed to be taken into consideration. In this paper, both the problems of overflow and quantization noise have been addressed adequately for the design process.

The remainder of this paper proceeds as follows. In section II, the conventional CORDIC algorithm is briefly reviewed.

Manuscript received March 31, 2013; revised May 22, 2013.

Amritakar Mandal and Rajesh Mishra are with the School of Information and Communication Technology, Gautam Buddha University, Greater Noida, U.P.- 201308 , India (e-mail: amritkar2k@gmail.com; rmishra@gbu.ac.in).

Design of pipelined CORDIC and design related issues have been discussed in section III. Evaluation and analysis of computational error has been discussed in section IV. In section V and VI, complex signal processing through CORDIC has been explained for demodulation purpose. In section VII, Hardware synthesis result has been discussed and conclusion may be found in section VIII.

II. Review of CORDIC Algorithm

The theory of CORDIC computation is to decompose the desired rotation angle into the weighted sum of a set of predefined elementary rotation angles. Each of them can be accomplished with simple shift-add operation for a desired rotational angle θ. It can be represented for M iterations of an input vector $(x,y)^T$ setting initial conditions: $x_0=x$, $y_0=y$, and $z_0=\theta$ as $z_f = \theta - \sum_{i=0}^{M-1} \delta_i\alpha_i$. If $z_f=0$ holds, then $\theta = \sum_{i=0}^{M-1} \delta_i\alpha_i$, *i.e.* the total accumulated rotation angle is equal to θ. δ_i, $0 \le i \le M-1$, denote a sequence of ± 1s that determine the direction of each elementary rotation. When M is the total number of elementary rotation angles, i-th angle α_i is given by:

$$\alpha_{m,i} = \frac{1}{\sqrt{m}}\tan^{-1}[\sqrt{m}\,2^{-s(m,i)}] = \begin{cases} 2^{-s(0,i)} \\ \tan^{-1} 2^{-s(1,i)} \\ \tanh^{-1} 2^{-s(-1,i)} \end{cases} \tag{1}$$

where m=0, 1 and -1 correspond to the rotation operation in linear, circular, and hyperbolic coordinate system respectively. For a given value of θ, the CORDIC iteration is given by:

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} 1 & -\delta_i 2^{-i} \\ \delta_i 2^{-i} & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} \tag{2}$$

and $z_{i+1} = z_i - \delta_i\alpha_i$

where $\alpha_i = \tan^{-1} 2^{-i}$. To bring a unit vector to desired angle θ, the CORDIC algorithm gives known recursive rotations to the vector. The known rotational values are shown in **Table. I** as a Pre-Computed angle. Once the vector is at desired angle, the outcome of the X and Y coordinates of the vector are equal to Cosθ and Sinθ respectively. Let a unit vector in iteration 'i' is rotated by some angle θ_i , then the recursively updated equations are generated in the following form:

$$\begin{aligned} x_{i+1} &= x_i \cos\delta_i\alpha_i - y_i \sin\delta_i\alpha_i \\ y_{i+1} &= y_i \cos\delta_i\alpha_i + x_i \sin\delta_i\alpha_i \end{aligned} \tag{3}$$

The above equation can be simplified and written as

$$\begin{aligned} x_{i+1} &= \cos\delta_i\alpha_i(x_i - y_i \tan\delta_i\alpha_i) \\ y_{i+1} &= \cos\delta_i\alpha_i(y_i + x_i \tan\delta_i\alpha_i) \end{aligned} \tag{4}$$

Here, $\tan\alpha_i$ is restricted to $\pm 2^{-i}$. So multiplication is converted in an arithmetic right shift. Since cosine is an even function, therefore $\cos(\alpha) = \cos(-\alpha)$. The iterative equation can be reduced to-

$$\begin{aligned} x_{i+1} &= K_i(x_i - y_i\delta_i 2^{-i}) \\ y_{i+1} &= K_i(y_i + x_i\delta_i 2^{-i}) \end{aligned} \tag{5}$$

Where $K_i = \cos\left(\arctan 2^{-i}\right) = \sqrt{(1+2^{-2i})}$ is known as gain factor for each iteration. If M iterations are performed, then scale factor, K, is defined as the multiplication of every K_i.

$$K = \prod_{i=0}^{M-1} K_i = \prod_{i=0}^{M-1} \sqrt{(1+2^{-2i})}\,. \tag{6}$$

The elementary functions sine and cosine can be computed using the rotation mode of the CORDIC algorithm if the initial vector starts at $(|K|,0)$ with unit length. The final outputs of the CORDIC for the given input values $x_0 = 1, y_0 = 0$ and $z_0 = \theta$ are as follows:

$x_f = K\cos\theta$, $y_f = K\sin\theta$ and $z_f = 0$.

Since the scale factor is constant for a given number of rotations, $x_0 = 1/K$ can be set to get purely $\sin\theta$ and $\cos\theta$ values.

Table I

Pre-Computed Angles for Pipelined CORDIC

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
..	…..	……..	…….

III. CORDIC Architecture

In this CORDIC architecture, a number of identical rotational modules have been incorporated and each module is responsible for one elementary rotation. Because of identical CORDIC iterations, it is convenient to map them into pipelined architecture [9]. The purpose of pipelined implementation is to device a minimum critical path. Therefore, this kind of architecture provides better throughput and lesser latency compared to other designs. It is associated with a number of stages of CORDIC Units where each of the pipelined stages consists of a basic CORDIC engine. The CORDIC engines are cascaded through intermediate latches as shown in **Fig. 1**. The shift operations are hardwired using permanent oblique bus connections to perform multiplications by 2^{-i} reducing a large silicon area as required by barrel shifters.The critical-path of the pipelined CORDIC is the time required by the Add/Substract operations in each of the stages. Every stage contributes critical path delay amounts to $T_{Path}=T_{Add} + T_{MUX} + T_{2C}$,where T_{Add}, T_{MUX} and T_{2C} are the time required for addition , 2:1 Multiplexing and 2's Complement operations, respectively. The pre-computed angles (**Table 1**) of i-th iteration angle α_i required at each CORDIC engine can be stored at a ROM memory location, are known. Therefore, the need of multiplexing and sign detection is avoided to reduce critical path. The latency of computation is thus depends primarily on the adder used. Since no sign detection is needed to force $z_f = 0$, the carry save adders are well suited in this architecture. The use of these adders reduces the stage delay significantly. The delay can be adjusted by using proper bit-length in the shift register. With the pipelining architecture, the propagation delay of the multiplier is the total delay of a single adder. So ultimately the throughput of the architecture is increased to a many fold as the throughput is given by: "1/delay due to a single adder". It implies that speed up factor becomes more than M and latency of the design is M times of the delay of a single adder. It is obvious that if we increase the number of iterations then the latency of the design also will increase significantly. If an iterative implementation of the CORDIC were used, the processor would take several clock cycles to give output for a given input. But in the pipelined architecture, it converts iterations into pipeline phases. Therefore, an output is obtained at every clock cycle after pipeline stage propagation. Each pipeline stage takes exactly one clock cycle to pass one output (Simulated output shown in **Fig. 2**).

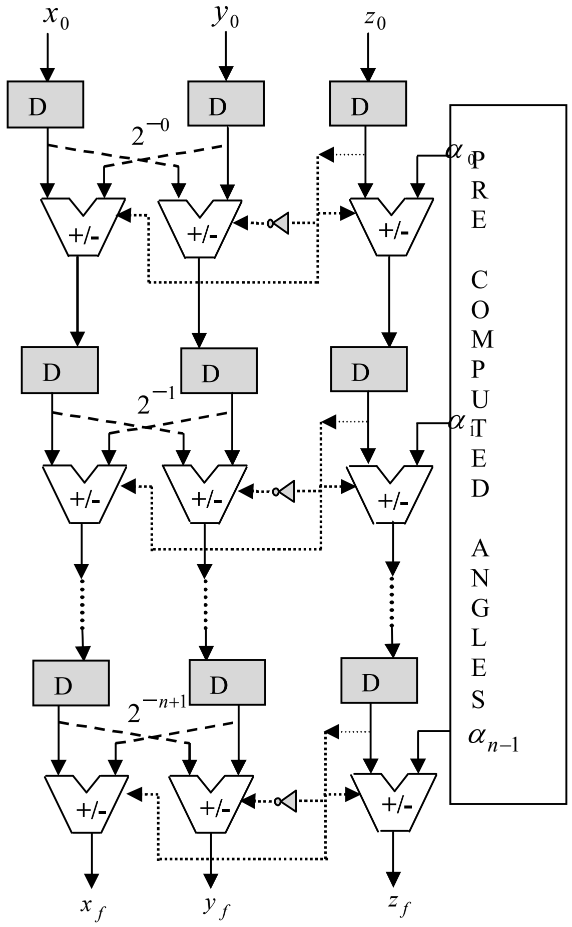

Fig. 1. Pipelined CORDIC Architecture

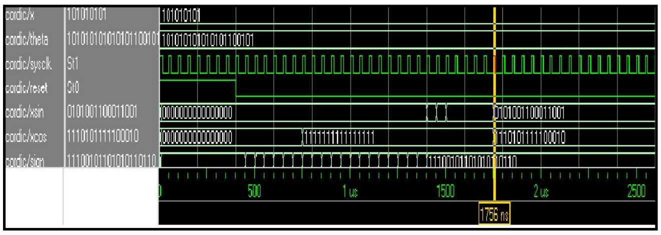

Fig. 2. Simulation Result of CORDIC

The most recurrent problems for a CORDIC implementation are overflow. Since the first tangent value is $2^0 = 1$, then rotation range will be $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$. The difference in binary representation between these two angles is one bit. Overflow arises when a rotational angle crosses a positive right angle to a negative one. To avoid overflow, an overflow control is added. It checks for the sign of the operands involve in addition or subtraction and the result of the operation. If overflow is produced, the result keeps its last sign without affecting the final result. In the overflow control, the sign of z_i determines whether addition or subtraction is to be performed.

IV. Numerical Error Analysis

Theoretically, CORDIC realization has infinite number of iterations and that leads to accurate result. But practically CORDIC realization uses finite number of iterations resulting in approximation error. This kind of error arises due to approximations in angle as well as finite word length [8]. To get total approximation error, the error due to the angle approximation process will be derived and followed by the error due to the truncation of word length will be derived. The total error is taken as the summation of the two.

In the angle approximation process the angle θ is approximated as the algebraic sum of predefined elementary angles.

$$\theta = \sum_{i=0}^{M-1} \sigma_i \alpha_i + \delta \qquad (7)$$

δ = Angle yet to be rotated after completion of the CORDIC iterations.

Due to convergence relationship of the CORDIC algorithm

$$\delta_{\max} = \tan^{-1} 2^{-(M-1)} \qquad (8)$$

Let v^* be the ideal result obtained by the rotation of the vector $v(0) = [x(0) \quad y(0)]^t$ by an angle of θ. Let $\tilde{v}(M)$ be the output of the CORDIC block after scaling operation and assuming that there is infinite precision in the CORDIC operation module. Then,

$$v^* = \begin{bmatrix} \cos\delta & -\sin\delta \\ \sin\delta & \cos\delta \end{bmatrix} . \tilde{v}(M) \qquad (9)$$

Let $D = \begin{bmatrix} \cos\delta & -\sin\delta \\ \sin\delta & \cos\delta \end{bmatrix}$

The error in the output due to the process of angle approximation is $v^* - \tilde{v}(M)$.

$$= v^* - \tilde{v}(M) = [D - I] * \tilde{v}(M) \qquad (10)$$

It can be easily shown that

$$\| D - I \| = \sqrt{(\cos\delta - 1)^2 + (\sin\delta)^2}$$
$$= 2 * \sin|\delta/2|$$
$$\| D - I \| = 2 * \sin|\delta/2| \le 2 * |\delta/2| = |\delta| \qquad (11)$$

From equation 9,

$$\| D - I \| \le |\delta| \le \tan^{-1}(2^{-(M-1)}) < 1/2^{M-1} \qquad (12)$$

we can get a consolidated truncation error due to finite wordlength using scale factor K and number of finite iterations (M).

$$K * \sqrt{2} * 2^{-b}(1 + \sum_{j=0}^{M-1}(\prod_{i=j}^{M-1}\sqrt{(1+2^{-2i}}))) \qquad (13)$$

The scaling operation also introduces some error which amounts to maximum of 2^{-b}. So the final expression for the total quantization error is addition of all previously mentioned errors.

|Total error| ≤

$$\frac{1}{2^{M-1}} * |v^*| + K * \sqrt{2} * 2^{-b}(1 + \sum_{j=0}^{M-1}\prod_{i=j}^{M-1}\sqrt{(1+2^{-2i}})) + 2^{-b} \qquad (14)$$

Let the output of the CORDIC block has 12 bits in its fractional part. Therefore, the upper limit of the total quantization error can be taken as 2^{-12}.

|Total error|

$$\le \frac{1}{2^{M-1}} * |v^*| + K * \sqrt{2} * 2^{-b}(1 + \sum_{j=0}^{M-1}\prod_{i=j}^{M-1}\sqrt{(1+2^{-2i}})) + 2^{-b} \le 2^{-12} \qquad (15)$$

The above inequality is simulated in MATLAB to find out fractional bits of the internal word length of the CORDIC.

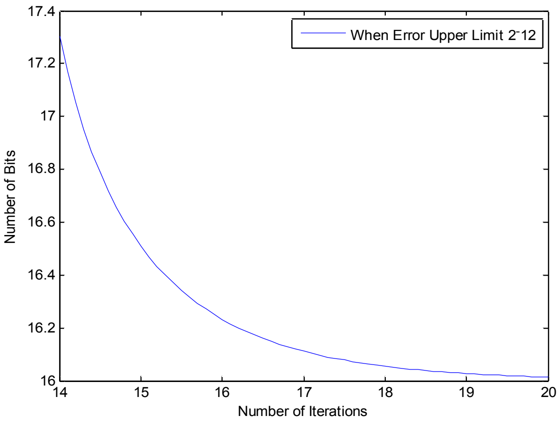

Fig. 3. The required Bits Vs. Iterations for CORDIC Internal Design

V. Basics of I/Q Demodulation

As per the Euler's theorem, vector sum of cosine 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 [11].

$$\cos\omega_c t + j\sin\omega_c t = e^{j\omega_c t} \qquad (16)$$

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) as shown in **Fig. 4**. If the carrier frequency of IF is f_c with a time varying amplitude $a(t)$ and time varying phase $\phi(t)$, then input signal $s(t)$ will be: $S(t) = a(t)\cos[(2\pi f_c t + \phi(t)]$. (17)

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 :

$$I(t) = a(t)\cos[(2\pi f_c t + \phi(t)].\cos(2\pi f_c t)$$
$$= a(t)\cos[\phi(t)] + a(t)\cos[4\pi f_c t + \phi(t)] \qquad (18)$$

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

$$I(t) = a(t)\cos[\phi(t)]. \qquad (19)$$

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

$$Q(t) = a(t)\sin[\phi(t)]. \qquad (20)$$

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

VI. Phase Detection TECHNIQUE in FMCW Radar

The down converted and filtered baseband signal has two components: real and imaginary parts. Therefore, the baseband 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 loop filter is sufficient for the digital demodulation using Digital PLL [10]. Using the vector rotation operator $[x, y]^T \angle\theta$, 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 : $\varepsilon_n = \Re\{v(n)\angle - \theta_n\}$

b). The loop filter equation: $c_n = 2\pi K_l \varepsilon_n$,where K_l is the loop filter coefficient. The loop filter coefficient K_l depends on the sampling frequency and number of

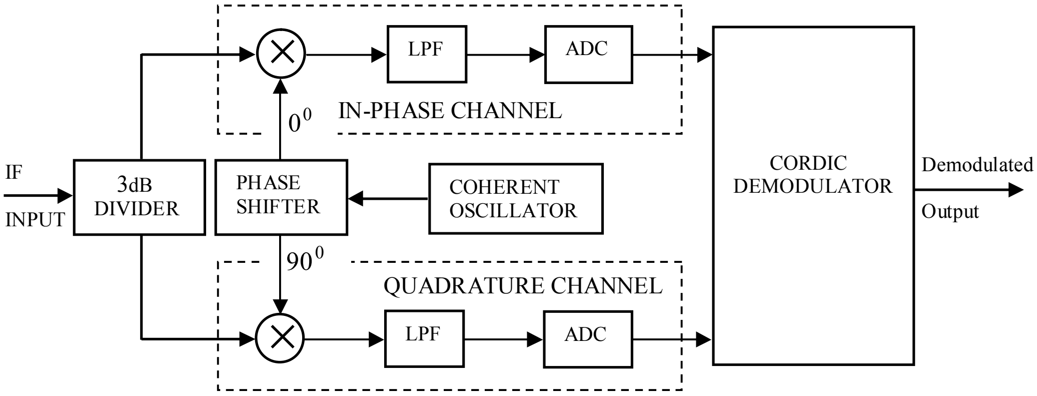

Fig. 4. CORDIC Based DPLL in I/Q Channel

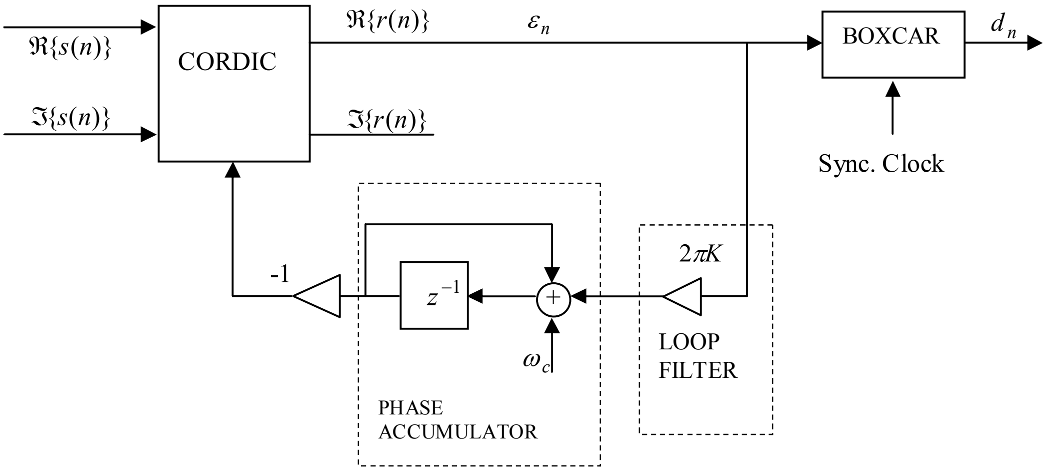

Fig. 5. Complex DPLL using CORDIC

iterations of CORDIC algorithm. For M number of iterations, the loop filter coefficient K_l can be given by:

$$K_{CORDIC} = \frac{K_l}{\prod_{i=0}^{M-1} \sqrt{1+2^{-2i}}} \qquad (21)$$

c). The phase accumulator equation:

$$\theta_{n+1} = (\theta_n + 2\pi K_l \varepsilon_n + \omega_c) \bmod 2\pi \,, \qquad (22)$$

where $\omega_c = 2\pi f_c$ is the center frequency.

The CORDIC based DPLL (as shown in **Fig. 5**) tries to adjust the continuous phase rotation in such a way that the complex component of the rotated vector will always be zero.

The post detection filter after the DPLL can be a Boxcar filter instead of low-pass filter. It rejects the unwanted noise Low-pass filter averages the signal and produces an output from the demodulated signal efficiently. BOXCAR filter has added advantage over the LPF as far as information recovery from narrow samples with little energy signal is concerned. Low-pass filter averages the signal and produces an output with weak amplitude signals. To avoid this problem, the samples can be stretched for entire inter sample period by increasing their sample amplitude at the filter output. The VLSI implementation of Boxcar generator is very easy as it performs only addition operation.

The simulation study has been carried out in MATLAB simulator. The down converted IF (with Doppler shift) is taken as input signal for the demodulator. The coherent oscillator frequency has been taken as a reference frequency to the demodulator. Any phase and amplitude variation (as shown in equations 18 and 19) in radar echo can be retrieved in the phase detector stage. Initial phase shift of 30^0 is introduced for the simulation. Simulated radar detected signal with reference to coherent reference signal and the final phase-amplitude variation at the demodulator output have been shown in **Fig. 6**, **Fig. 7** and **Fig. 8** respectively.

VII. Hardware Synthesis Result

The main part of VLSI design is optimization of design in terms of speed, power, resource utilization and delay etc. The

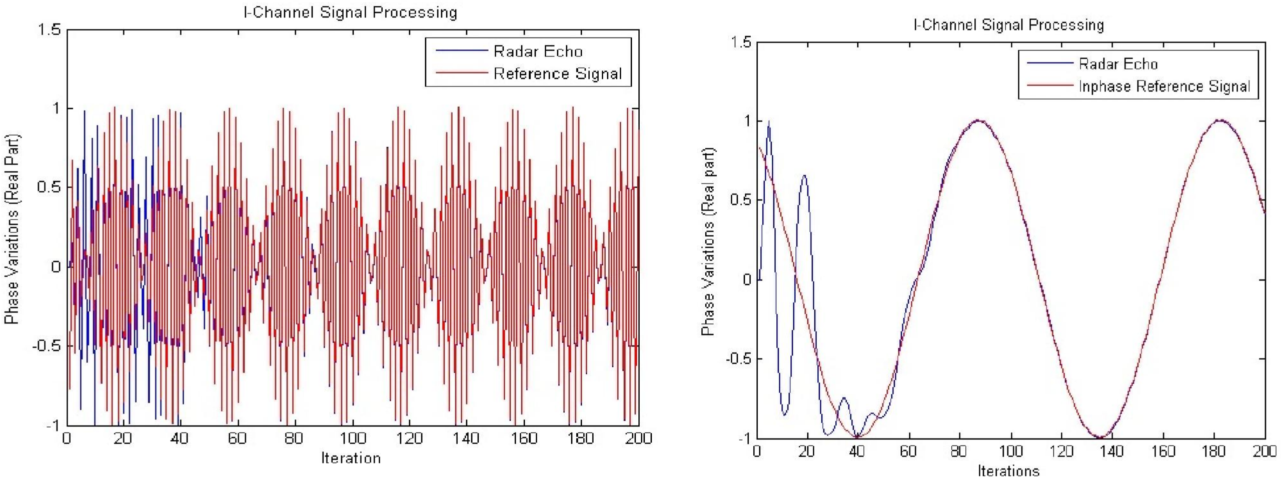

Fig. 6. Phase variation in In-phase Channel Radar Receiver

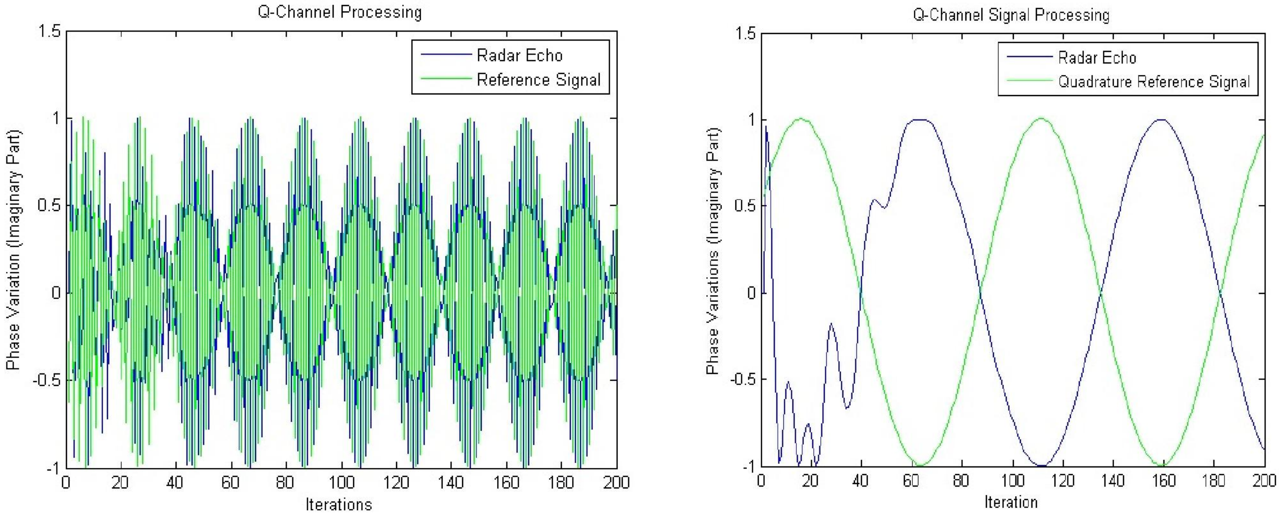

Fig. 7. Phase variation in Quadrature Channel Radar receiver

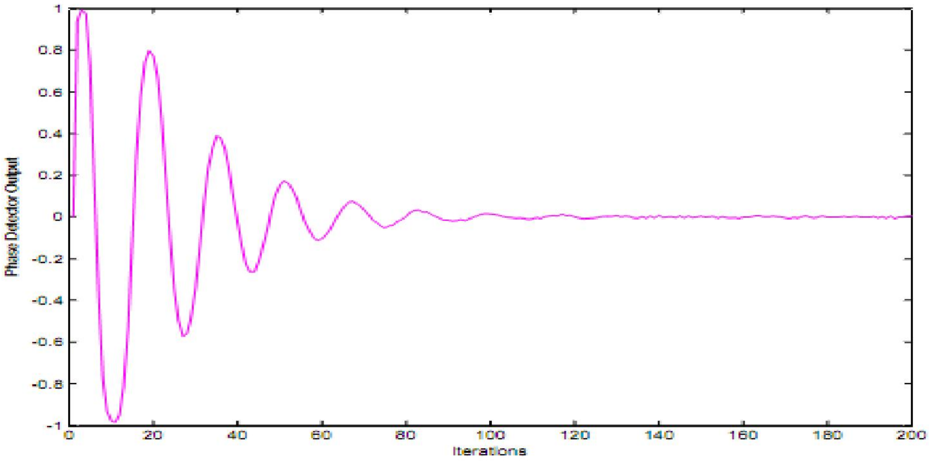

Fig. 8. Phase Detected Output at Radar Receiver

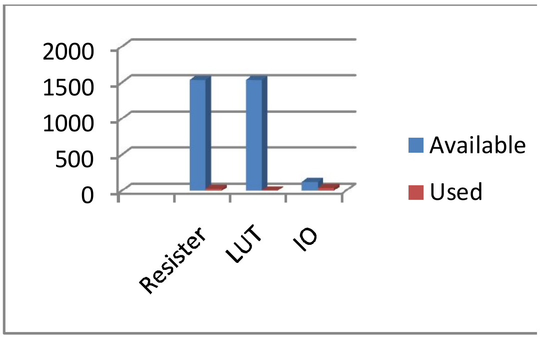

Fig. 9. Resource Utilization

proposed architecture design was synthesized on Spartan-3 based xc3s50pq208-5 FPGA device using XILINX ISE 10.1 and simulated on ModelSim. The area utilization of proposed design is implemented on above said FPGA kit in terms of Register, LUT and IOs. The area consumed by register, LUTs and IOs are 3%, 2% and 35% respectively of available resources as shown in **Fig. 9**.The result of power measurement for proposed architecture also has been measured using Cadence custom IC design tool and shown in **Table II**.

Table II.

Power Measurement

Cells	**Leakage Power (nW)**	**Dynamic Power (µW)**	**Total Power (µW)**
95	112.132	112.64	112.752

VIII. Conclusion

The paper presents the demodulation technique in a high speed FMCW Radar receiver using complex Digital Phase Locked Loop. To maintain high degree of accuracy during radar signal processing, quantization error is needed to be minimized in design level itself. For that purpose, numerical error analysis has been done to minimize angle approximation as well as truncation errors. With using reduced number of micro-rotation and adequate optimized convergence property of CORDIC design, implementation of this kind of demodulator becomes easier. Inherent issue of overflow is quite appropriately resolved in the proposed design. Numbers of micro-rotations have been adjusted so as to achieve better loop performance and speed of operation while minimizing quantization error. The phase and amplitude variations in the received echo signal are aptly demodulated. The BOXCAR filter gives additional facility in reconstructing output signal efficiently.

References

[1] W. Xu, C. Gu, C. Li, M. Sarrafzadeh, "Robust doppler radar demodulation via compressed sensing", Electronics Letters, vol. 48, no. 22, pp. 1428-1430, Oct 2012.

[2] N. Da Dalt, "Linearized Analysis of a Digital Bang-Bang PLL and its Validity Limits Applied to Jitter Transfer and Jitter Generation", *IEEE Transactions on Circuits and Systems I:Regular paper*, vol. 55, no. 11, pp. 3663-3675, Dec. 2008.

[3] J. E. Volder, "The CORDIC Trigonometric Computing Technique", *IRE Transactions on Electronic Computing*, vol. EC-8, pp. 330-334, Sept, 1959.

[4] Y. H. Hu, "CORDIC-Based VLSI Architectures for Digital Signal Processing", *IEEE Signal Processing Magazine*, vol. 9, no. 3, pp. 16-35, 1992.

[5] S. Ugazio, " Design of real time adaptive DPLLs for generic and variable Doppler frequency", *In proc. Of International Conference on Localization and GNNS (ICL-GNSS)*", pp. 169-174, Tampere, Jun 2011.

[6] K. Kota and J.R. Cavallaro, "Numerical accuracy and hardware trade-offs for CORDIC arithmetic for special-purpose processors," IEEE Trans. Comput., vol.42, no.7, pp.769–779, Jul 93.

[7] Y. H. Hu, "The Quantization Effects of the CORDIC Algorithm", *IEEE Transactions on signal processing,* vol. 40, no. 4, pp. 834-844, Apr 1992.

[8] S.Y. Park and N.I. Cho, "Fixed-point error analysis of CORDIC processor based on the variance propagation formula," *IEEE Trans. Circuits Syst. I*, vol.51, no.3, pp.573–584, March 2004.

[9] T. –Y. SUNG, H. –C. HSIN, "Fixed-Point Error Analysis of CORDIC Arithmetic for Special-Purpose Signal Processors", *IEICE Trans. Fundamentals*, vol. E90-A, no. 9, Sept 2007.

[10] J. Vuori, "Implementation of a Digital Phase-Locked Loop Using CORDIC Algorithm", *IEEE International Symposium on Circuits and Systems,* vol. 4, pp. 164-167, Atlanta, 1996.

[11] B. R. Mahafza, A. Elsherbeni, "MATLAB Simulations for Radar Systems Design", CRC, Florida, USA, 2003.

Amritakar Mandal received his Bachelor of Engineering Degree in Electronics and Communication Engineering from The Institution of Engineers (India) in 2006. He completed M.Tech with specialization in VLSI from Shobhit University, India in 2009. He is currently working toward the Ph.D Degree in Electronics and Communication Engineering in the Department of Information and Communication Technology, Gautam Buddha University, India. He has vast experience in Electronic Warfare and Surveillance Radar systems. His research interests include Radar Signal Processing and High Speed VLSI Design for Communication Systems.

Rajesh Mishra is currently working as Assistant Professor in School of Information and Communication Technology, Gautam Buddha University Greater Noida, Delhi NCR (India). He received his BE (Electronics Eng.), M. Tech and Ph. D. degree (Reliability Eng.) from Reliability Engineering Centre, IIT Kharagpur (India) in year 2000, 2004, and 2009 respectively. He has research interest in the area of reliability engineering, layout design for capacitated networks, and network optimization. He has published papers in several international journals such as IJPE, IEEE, RESS, QTQM etc.