International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 8, August - 2013

Implementation of Frequency Down Converter using CORDIC Algorithm on FPGA

Yogendra Kr. Upadhyaya [#1]
Electronics and Communication Engineering
National Institute of Technology
Kurukshetra
Haryana, INDIA

Dr. R. K. Sharma [*2]
Electronics and Communication Engineering
National Institute of Technology
Kurukshetra
Haryana, INDIA

Abstract—In a communication system, the received signals are of high data rates making it difficult to process the signals to extract information of interest. So to solve this problem DDC makes a better solution. In this efficient way of designing and implementing a Wideband Digital down converter has been discussed. Though the received signal is RF signal with high data rates, IF stage is used to frequency shift the signal to fixed IF which is input to ADC. This is sampled and given as input as input to DDC.CORDIC generator is used instead of numerically controlled oscillator (NCO).
It is shown that filter bandwidth varies by decimation factor. Decimation range in this paper is 2 to 16384. Filtering is implemented in stages to obtain efficient response. Xilinx 13.2 version is used to simulate each block of DDC at system level testing and Sparten-3 FPGA with speed -4 is hardware used implementing the design.

***Keywords*—Wideband Digital down converter, ADC, Baseband signal, Decimation CORDIC generator, FPGA, System level, Board level testing.**

I. INTRODUCTION

Communication plays vital part in day to day life for transfer of information. Though there are different modes of communication at present Digital Communication is more popular. It is a process of transferring signals, in digital format i.e. as bits. A transmitter, channel and receiver are the main blocks of communication system. The DDC presented in this paper is the key component of Receiver. The digital IF signal from ADC depends on the band of interest as the Nyquist's theory states that signal should be sampled at rate "at least double the bandwidth of interest". A DDC allows the frequency band of interest to be moved down the spectrum to baseband signal near to 0 HZ such that further processing on signals become easier. Later techniques are involved for varying the filter specifications to extract the signal of interest.

The remainder of this paper is structured as follows: Section 2 provides an overview of Digital down Converter. Section 3 mentions the steps involved in converting IF signal to base band signal. Section 4 is about filtering and decimation. Section 5 gives a detailed explanation on implementing the design on FPGA. Section 6 provides Simulation results and the last section.

II. OVERVIEW OF DDC

Down Conversion involves the process of the shifting a high rated signal to a standard signal. Generally the receivers receive wide band of signals but end user may only require a small portion of the entire band. So fulfilling the above requirement might involve prohibitively large filters. A variable decimation DDC makes this process easier.

A DDC consist of four basic blocks

I. CORDIC(Coordinate Rotation by Digital Computer)
II. Mixer
III. VARCIC(Variable Cascaded Integrated Comb) filter
IV. CIC (Cascaded integrated Comb) filter
V. FIR(Finite Impulse Response) filter

A. CORDIC Algorithm:-The CORDIC (Coordinate Rotation Digital Computer) was developed by Jack Volder in 1959 [1] as an iterative algorithm to convert between polar and Cartesian Coordinates using shift, add and subtract operations only. It can be implemented with shift-add/subtract type algorithm. It can also be used to compute trigonometric functions. Examples sine, cosine, polar to rectangular coordinates etc.

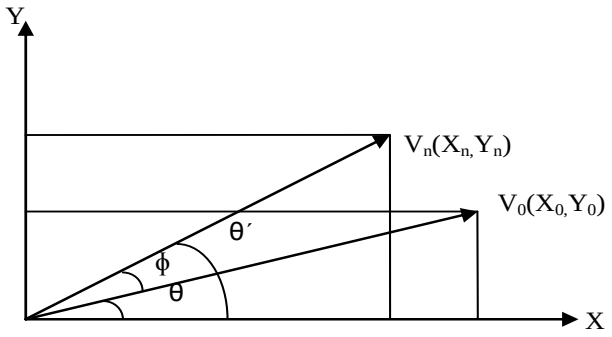


Fig.1 Vector V rotating with phase ϕ

In circular rotation made a CORDIC function could compute the Cartesian Coordinates of the target vector V_n by rotating the input vector V_0 by an arbitrary angle $\phi=z_0$. so how do we calculate X_n and Y_n based on input vector and angle

$$X_n = X_0 \cos\phi - Y_0 \sin\phi$$

$$Y_n = Y_0 cos\phi + X_0 sin\phi$$
$$X_n = cos\phi(X_0 - Y_0 tan\phi)$$
$$Y_n = cos\phi(Y_0 + X_0 tan\phi)$$

So far, nothing is simplified however if rotation angles are restricted so that tanϕ =,the multiplication by tangent term is reduced to simple shift operation.tanϕ =$\pm 2^0$,2^{-1}, 2^{-2},2^{-3}

Table:-1 for 8-bit CORDIC Hardware

i	$d^{-i}=2^{-i}$= tanϕ$_i$	Φi=arctan(2^{-i})	Φ$_i$in radains
0	1	45^0	0.7854
1	0.5	26.565^0	0.4634
2	0.25	14.036^0	0.245
3	0.125	7.125^0	0.1244
4	0.0625	3.576^0	0.0624
5	0.3125	1.7876^0	0.0312
6	0.015625	0.8938^0	0.0156
7	0.0078125	0.4469^0	0.0078

But what if my desired angle of rotation is not exactly one of these values. The desired angle of rotation is obtainable by performing a series of successively smaller elementary rotations where i=0--------n-1 for tanϕ = $\pm 2^{-i}$

Let say our desired rotation is 30 degrees.

Table:-2 choosing the signs of the rotation angles to force z to zero

i	z_i-α_i	z_{i+1}
0	30.0-45.0	-15
1	-15+26.6	11.6
2	11.6-14.0	-2.4
3	-2.4+7.1	4.7
4	4.7-3.6	1.1
5	1.1-1.8	-0.7
6	-0.7+0.9	0.2
7	0.2-0.4	-0.2
8	-0.2+0.2	0
9	0.0-0.1	-0.1

We start at iteration 0 with an angle of $Z_0=30^0$.If the angle $Z_i>0$ then we subtract the tan(ϕ) angle, otherwise we add the tan(ϕ) angle and make our approximate X_i and Y_i calculations.

$$X_n = cos\phi(X_0 - Y_0 tan\phi)$$
$$Y_n = cos\phi(Y_0 + X_0 tan\phi)$$

If the decision at each iteration i, is which direction to rotate rather than whether or not to rotate then the cos(ϕ_i) term becomes a constant because cos(ϕ_i)=cos(-ϕ_i). In other words the cos (ϕ_i) is not dependent on direction of rotation. The iterative rotation can now be expressed as:

$$X_{i+1}=K_i [X_i - d_i Y_i 2^{-i}]$$
$$Y_{i+1}=K_i [Y_i + d_i X_i 2^{-i}]$$
$$K_i=cos(tan^{-1}(2^{-i}))$$
$$K_i=\frac{1}{\sqrt{(1+2^{-2i})}}$$
$$d_i=\pm 1$$

Removing the scale constant from iterative equations yields a shift-add algorithm for vector rotation. The product of K_i's can be applied elsewhere in system or treated as part of a system processing gain. The product approaches 0.6073 as the number of iterations goes to infinity. Therefore, the rotation algorithm has A_n, of approximately 1.647. The exact gain depends on the number of iterations and obeys the relation

$$A_n=\prod\sqrt{1+2^{-2i}}$$

It also needs to be noted that the previous equations are valid for rotation angles between

$$-\pi/2 \le \phi \le \pi/2$$

In order to increase the convergence range for all rotation angles $|Z_0|<\pi$ volder proposed an intial iteration which rotates the input vector by $\pm\pi/2$

$$X_0' = -d.Y_0$$
$$Y_0' = d.X_0$$
$$Z_0' = Z_0 - d.\frac{\pi}{2}$$

Where $d=\begin{cases}+1 \text{ if } Z_0<0\\ -1 \text{ otherwise}\end{cases}$

The elementary angles can be expressed in any convenient angular unit. Those angular values are supplied by small look-up table (one entry per iteration) or they are hardwired depending on implementation. The angle accumulator adds a third difference equation to CORDIC algorithm.

$$Z_{i+1}=Z_i - d_i tan^{-1}(2^{-i})$$

Thus we now have a set of accumulation equations to use for each iteration[2]

$$X_{i+1}=K_i [X_i - d_i Y_i 2^{-i}]$$
$$Y_{i+1}=K_i [Y_i + d_i X_i 2^{-i}]$$
$$Z_{i+1}=Z_i - d_i tan^{-1}(2^{-i})$$

Where $d=\begin{cases}+1 \text{ if } Z_0<0\\ -1 \text{ otherwise}\end{cases}$

Where d_i determines the direction for each elementary rotation. Therefore after n iterations the CORDIC equations

$$X_n = A_n [X_0 cos(z_0) - Y_0 sin(z_0)]$$
$$Y_n = A_n [Y_0 cos(z_0) + X_0 sin(z_0)]$$
$$Z_n = 0$$

Where A_n is originally expected gain.

Therefore a CORDIC function could be used to rotate vectors.We define a vector where $X_0=0$

$$X_n \approx -A_n Y_0 sin(z_0)$$
$$Y_n \approx A_n + X_0 sin(z_0)$$

By selecting Y_0 equal to $1/A_n$, the rotation produces the unscaled sine and cosine of angle argument z_0, very often the sine and cosine module a magnitude value. Using other techniques such a sin0/cos0 look-up table requires a pair of multipliers to obtain the modulation. The CORDIC technique performs the multiply as part of the rotation operation and therefore eliminates the need for a pair of explicit multipliers.

III. MIXER

Signal modulation involves changes made to sine waves in order to encode information. The mathematical equation representing a sine wave is as follows

$$S_{BP}(t) = A cos(2\pi ft+\phi)$$

Where A is amplitude, f is frequency and ϕ is phase.

$$S_{BP}(t) = A\cos(2\pi ft+\phi)$$

$$S_{BP}(t)=A\cos(2\pi ft)\cos\phi - A\sin(2\pi ft)\sin\phi$$

$$If \quad I= A\cos\phi$$

$$Q=A\sin\phi$$

$$A\cos(2\pi ft+\phi) = I\cos(2\pi ft) - Q\sin(2\pi ft)$$

Where I is the amplitude of in phase carrier

Q is the amplitude of quadrature-phase carrier.

Remember that difference between a sine wave and cosine wave of the same frequency of is 90-degree phase offset between them. The implication of this are very important what this essentially means is that we can control the amplitude, frequency and phase of modulating RF carrier sine wave. We can achieve the same effect by manipulating the amplitudes of input I and Q Signals. Of course the second half of equation is a sine wave and first half is a cosine wave so we must include a device in the Hardware circuit to induce a 90 degree phase between the carriers but this is a much simpler design issue than direct phase manipulation.

If:

$I = A\cos\phi$

$Q = A\sin\phi$

A. CIC Filter

The cascaded integrator-comb (CIC) filter is a class of hardware-efficient linear phase finite impulse response (FIR) digital filters [6]. The CIC filter is suitable for this high-speed application because of its ability to achieve high decimation factors and other reason is it is implemented using additions and subtractions rather than using multipliers. It decimates by R which is programmable. The two basic building blocks of a CIC filter are

1) An integrator (decimator): An integrator is simply a single-pole IIR filter with a unity feedback coefficient [9],[10]

$$y[n] = y[n-1] + x[n]$$

This system is also known as an accumulator [9],[10]. The transfer function for an integrator on the z-plane is

$$H_I(z) = 1/(1 - z^{-1})$$

2) Comb Filter (Interpolator): A comb filter running at the slow sampling rate fs/R is described by

$$y[n] = x[n] - x[n-D].$$

A comb filter is a differentiator with a transfer function

$$H_C(z) = (1 - z^{-D})$$

In this equation, *M* is the differential delay, and is usually limited to 1 or 2. To summarize, a CIC filter would have N cascaded integrator stages clocked at fs, followed by a rate change by a factor R, followed by N cascaded comb stages running at fs/R [10]

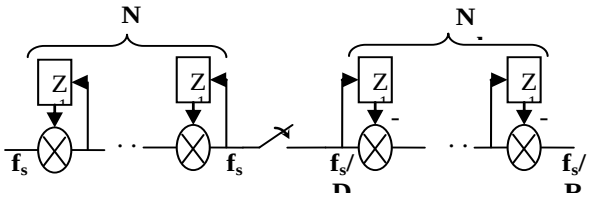


Fig. 2 CIC Filter

Each integrator contributes to the CIC transfer function with a pole. Each comb section contributes with a zero of order D, where D is the frequency decimation ratio. The CIC transfer function in the Z-plane becomes:

Frequency Characteristics:

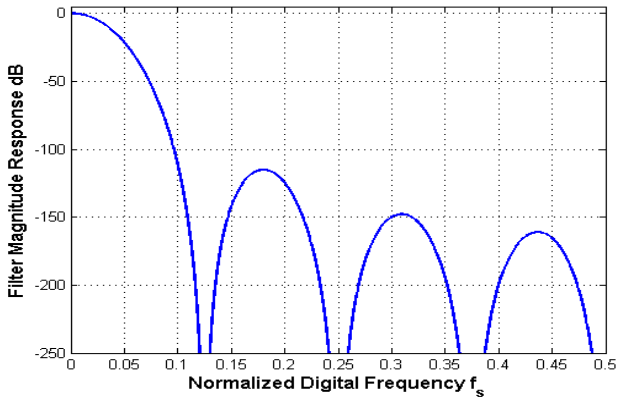
The transfer function for a CIC filter at fs is

$$H(z) = H_I^N(z)\,H_C^N(z) = \frac{\left(1 - z^{-D}\right)^N}{\left(1 - z^{-1}\right)^N} = \left[\sum_{k=0}^{D-1} z^{-k}\right]^N$$

We must be careful here because we have two sampling frequencies in the system, related by D. If we evaluate the z-transference at the output sampling frequency $z=\exp(j2\pi f_s/D)$,The magnitude response at the output of the filter is as shown below[3] .We can obtain an expression for the CIC filter's frequency response by evaluating $H_{cic}(z)$ transfer function on the *z*-plane's unit circle, by setting $z = e^{j2\pi f}$, yielding a sinc like function.

As already mentioned the frequency response of CIC filters is affected by the parameters N, M, R. Differential delay, M, affects the location of nulls at any given rate change value and increases attenuation levels generally at all lobes in the response. Varying the rate change value, R, adjusts the null positions up or down accordingly without having much affect on the attenuation of each lobe and increasing the number of stages increases attenuation of the lobes without shifting null positions.

$$H(f) = \left[\frac{\sin(\pi f)}{\sin(\pi f / D)}\right]^N \approx \left[D.\frac{\sin(\pi f)}{\pi f}\right]^N \quad for \quad D >> 1$$



Compensation FIR filter:

The output of the CIC filter has a sinc shape, which is not suitable for most applications. A "clean-up" filter can be applied at the CIC output to correct for the pass band droop, as well as to achieve the desired cut-off frequency and filter shape.This filter typically decimates by a factor of 2 or 4 to minimizethe output sample [3],[10].

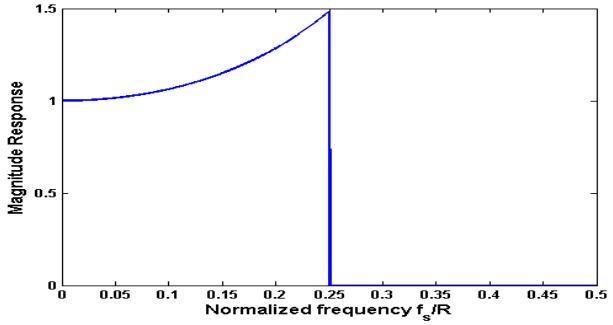


Fig. 4 Magnitude Response of CFIR Filter

IV. Simulation Results:-

CORDIC Based Generator

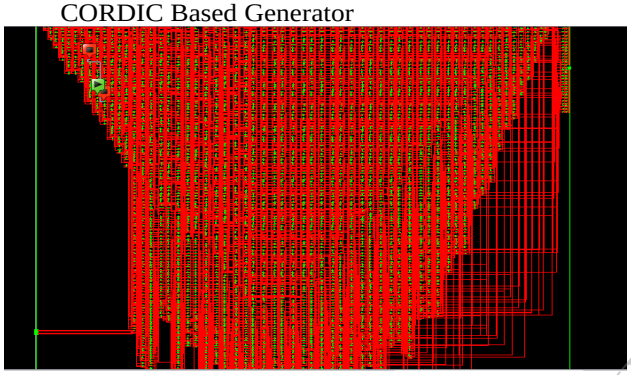


Fig.5 RTL Schematic view of CORDIC

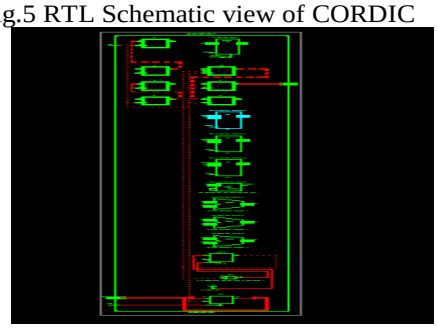


Fig.6 RTL Schematic view of CIC Filter

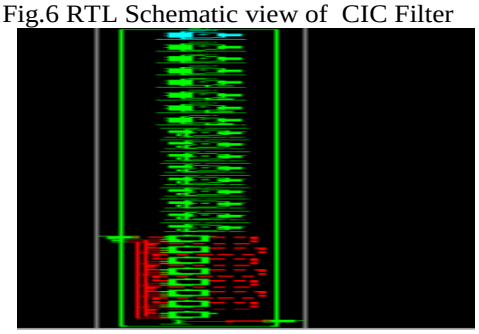


Fig.7 RTL Schematic view of FIR Filter

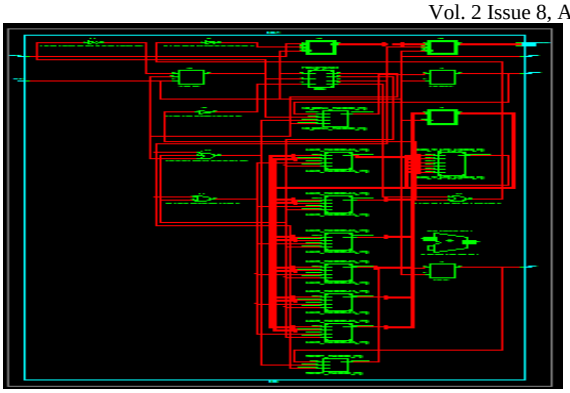


Fig.8 RTL Schematic view of CORDIC-DDC

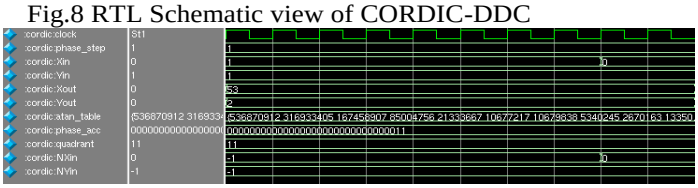


Fig.9 Output for input Xin=1 & Yin=1 in quadrant I

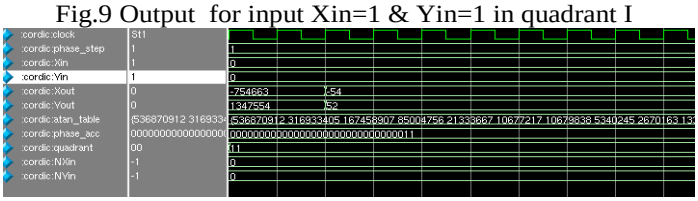


Fig.10 Output for input Xin=0 & Yin=0 in quadrant I

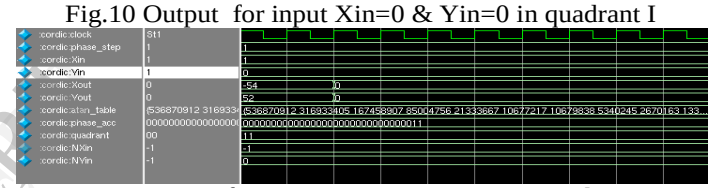


Fig.11 Output for input Xin=1 & Yin=0 in quadrant I

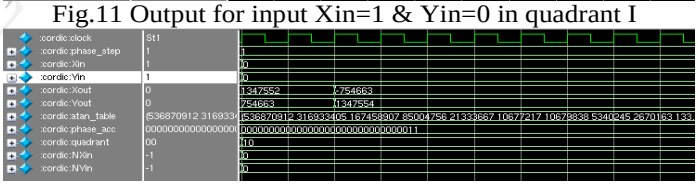


Fig.12 Output for input Xin=1 & Yin=0 in quadrant II

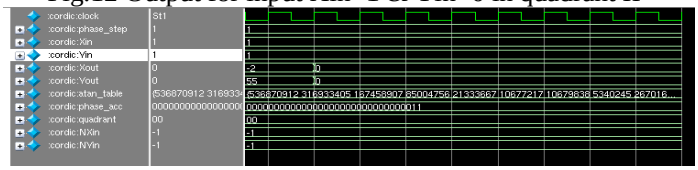


Fig.13 Output for input Xin=1 & Yin=1 in quadrant III

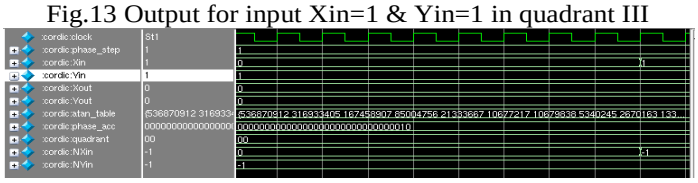


Fig.14 Output for input Xin=0 & Yin=1 in quadrant III

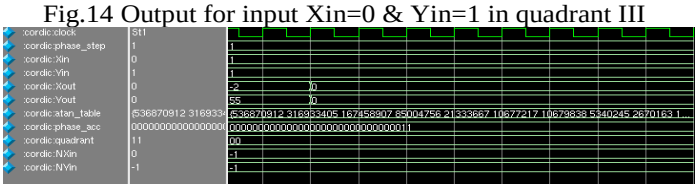


Fig.15 Output for input Xin=1 & Yin=1 in quadrant III

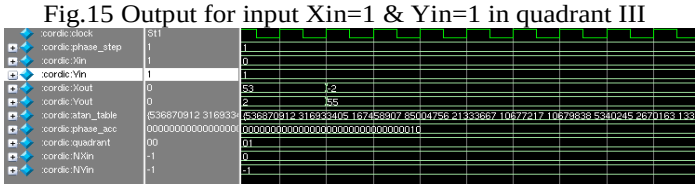


Fig.16 Output for input Xin=0 & Yin=1 in quadrant IV

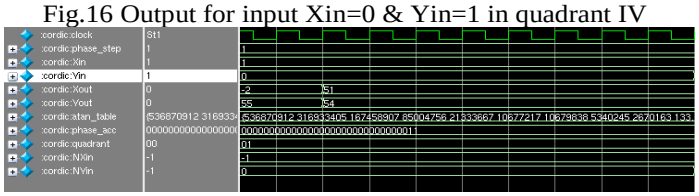


Fig.17 Output for input Xin=1 & Yin=0 in quadrant IV

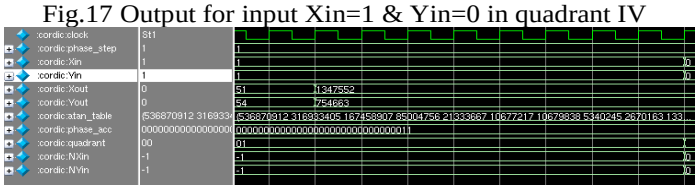


Fig.18 Output for input Xin=1 & Yin=1 in quadrant IV

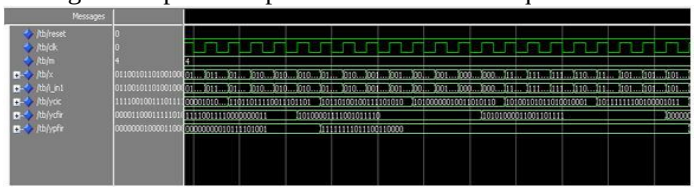


Fig.19 Output of CIC filter for different inputs

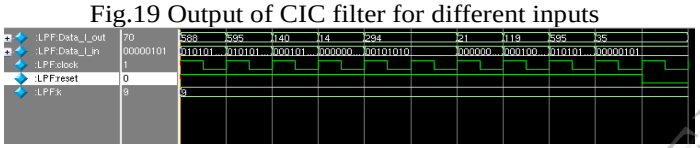


Fig.20 Output of Low Pass Filer

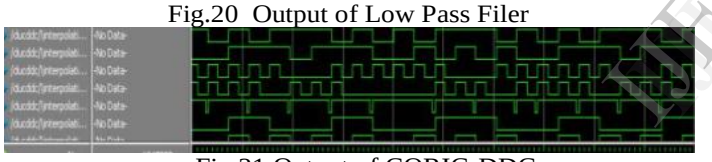


Fig.21 Output of CORIC-DDC

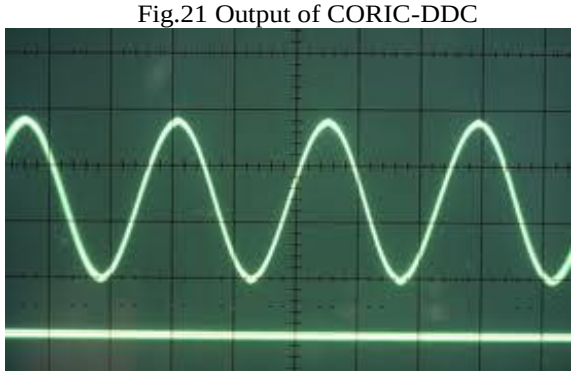


Fig.22 Sine wave on CRO through FPGA Sparatn3E

CORDIC-DDC:- An approach to overcome this drawback is the calculation of the corresponding sin0/cos0 values by means of CORDIC with the main advantage of using only a small look up table(~n×n bit).The major drawback of CORDIC approach is increased circuit complexity. However it used in the context of digit down conversion of frequency synchronization, the additional hardware effort is partly compensated because there is no need for explicit multiplier. This is where our previously shown CORDIC sin0/cos0 functions come in

$$X_n = A_n Y_0 \sin(z_0)$$

$$Y_n = A_n + X_0 \sin(z_0)$$

Where X_0 set to 0.This will allow to create a CORDIC-DDC.

At each clock interval,Y_0 is loaded with current samples $S_{BP}(k)$ of an IF input signal and Z_0 with the current sample $\phi(k)$.The latter is supplied by an overflowing phase accumulator which generates the oscillator frequency f_0.X_0 is set to zero. After n+1 iterations the CORDIC provides the samples I(k) and Q(k) of down converted in phase and quadrature-phase signal with a resolution of approximately n bits. In order to achieve this, only a very small look-up table is needed. It contain the n+2 basic rotation angles. Still the main problem of the CORDIC is that (n+1) iterations have to be performed for each signal sample requiring an internal clock rate being (n+1) times higher than sample rate of the signal. However the CORDIC can be implemented by pipelined arithecture which eliminates the n+1 factor. Thus the CORDIC-DDC becomes suitable for the high –speed applications.

An additional advantage of such an implementation is that there is no need for look-up table anymore, since the invariant elementary rotation angles can be hardwired to each pipeline stage. In other words a specific $\tan^{-1}(\phi)$ will always be used for each pipeline stage.

V. CONCLUSION:

The CORDIC algorithm is powerful and widely used tool for digital signal processing applications and can be implemented using PDPs (Programmable Digital Processor) But large amount of data processing is required because of complex computations. This affects the cost, speed and flexibility of DSP system. So the implementation of DDC (Digital Down Converter) using CORDIC algorithm on FPGA is need of day as the FPGA can give enhanced speed at low cost with lot of flexibility. This is due to the fact that hardware implementation of a lot of multipliers can be done on FPGA with are limited in case of PDPs. It can be concluded that the desiged.RTL model for CORDIC and DDC is accurate and can work for real time applications.

The number resolution of frequency increases the size of the ROM/LUT for CORDIC increases exponentially. This reduces the speed of DDC and increases the size of hardware. This issue of increase in the size of ROM/LUT can be solved by using memory reduction techniques like phase truncation and quadrature symmetry of sine wave. In our design we have implemented a 12 bit DAC on SPARTAN 3E board and analyze the effect of frequency using CORDIC algorithm.

vii. FUTURE SCOPE

CORDIC algorithm is implemented in large FFT instead of Booth Multipliers, OFDM system, in DCT and in DWT using Pipelined-Parallel algorithm. CORDIC-DDC is implemented in SDR (Software Defined Radio) and beam loss accounting system.

REFERENCES

[1] J.E.Volder: The CORDIC trigonometric comp. technique.IRE Trans. Elec. Comp.(1959), vol. EC-8-3, pp. 330-334.

[2] Ray Andraka: A survey of CORDIC algorithms for FPGAs .Proceedings of the 1998 ACM/SIGDA 6th international symposium on FPGAs, Monterey, CA (1998), pp191-200.

[3] GC4016 Multistandard Quad DDC Chip Data Sheet, Rev. 1.0. August 2001, Texas Instruments. (Formerly Graychip Inc.). Document: slws133a.pdf.

[4] Tjerk Bijlsma, Pascal T. Wolkotte, Gerard J.M. Smit "An Optimal Architecture for a DDC" 2006 IEEE.

[5] Stephen Creaney and Igor Kostarnov "Designing Efficient Digital Up and Down Converters for Wide band Systems". XAPP1113 (v1.0) November 21, 2008.

[6] T.Hollis / R.Weir, "Theory of Digital Down Conversion", Hunt Engineering, 2003.Rev 1.2.

[7] Xilinx "DDS Compiler v2.1", Product specification DS5558 March 21,2008.

[8] E. B.Hogenauer. An economical class of digital filters for decimation and interpolation. IEEE Transactions on Acoustics, Speech and Signal Processing, ASSP-29(2):155{162, 1981.

[9] Alan Y. Kwentus, Zhongnong Jiang, and Alan N. Willson. Application of Filter Sharpening to Cascaded Integrator-Comb Decimation Filters. IEEE Transactions on Signal Processing, Vol. 45, No. 2, February 1997 457.

[10] Altera's application note 455 April 2007, "Understanding CIC compensation filter" ver. 1.0.

[11] Alan V. Oppenheim and Ronald W. Schafer. Discrete-Time Signal Processing. Prentice-Hall Signal Processing Series. Prentice-Hall, Englewood Cliffs, 1989.

[12] IEEE Transactions on Acoustics, Speech and Signal Processing,Vol. ASSP-23, no. 5 , October 1975 Optimum FIR Digital Filter Implementations for Decimation, Interpolation, and Narrow-Band Filtering.

[13] R. R. Shively, "On multistage FIR filters with decimation," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-23, pp. 353-357, Aug. 1975.

[14] Advanced Digital Design with Verilog HDL PHI Learing Private Limited 3rd edition, 2005.

[15] Field programmable gate array, S. Brown, R.J.Francis, J.Rose ,Z.G.Vranesic, 2007, BSP.