ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 2, Issue 4, July 2013

# Design, Implementation and Simulation of Adaptive FIR Filter using CORDIC Structures for Radar Applications

Vanitha.A[1], Venkatesh Kumar N[2]
[1,2]Electronics and Communication department, Jain University
Jakkasandra Post, Kanankapura Taluk Bangalore Rural district-562112

***Abstract*— The design of high speed VLSI architectures for real-time digital signal processing (DSP) algorithms and their realizations has been directed by the advances in the VLSI technology. This paper presents the implementation of CORDIC algorithm for adaptive signal processing by using pipelined Coordinate Rotation Digital Computer (CORDIC) unit and pipelined multiplier to get high system throughput in each of the pipelined stage. CORDIC algorithm is extremely useful in efficient and effective implementation of DSP systems. This algorithm allows implementation of trigonometric functions like sine, cosine, magnitude and phase with great precision by using just simple shift and adding operations. 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. The area is reduced by optimizing the number of micro rotation and number of iteration to minimize quantization error.***

*Index Terms*— **CORDIC, Pipelined architecture, Micro-rotation, Quantization error.**

## I. INTRODUCTION

The coordinate rotation digital computer (CORDIC) has established its popularity in several important areas of application, like generation of sine and cosine functions, calculation of discrete sinusoidal transforms like fast Fourier transform (FFT), discrete sine/cosine transforms (DST/DCT), householder transform (HT)[14]. Many variations have been suggested for efficient implementation of CORDIC with less number of iterations over the conventional CORDIC algorithm .The number of CORDIC iterations are optimized by greedy search at the cost of additional area and time for the implementation of variable scale-factor. Efficient scale-factor compensation techniques are proposed, which adversely affect the latency/throughput of computation [2]. The enhanced scale-free CORDIC in combines few conventional CORDIC iterations with scaling-free CORDIC iterations for an efficient pipelined CORDIC implementation with improved RoC. However, if used for recursive CORDIC architecture, combining two different types of CORDIC iterations degrades performance. The proposed architecture has comparable or less area complexity with other existing scaling-free CORDIC algorithms.
This paper designs an adaptive FIR filter based Moving Target Detector (MTD). In MTD, an adjustable local sine/cosine wave generator is required [8]. 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.

## II. BRIEF OVER VIEW OF CORDIC ALGORITHM

In this section necessary background for the work such as Taylor series, Lookup table and CORDIC are discussed to evaluate trigonometric functions.

### *A. Taylor Series*

The Taylor series expansion for sine is:

$$\text{Sin}(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \text{.....}$$

This method is one of the oldest and most widely, but the problem associated with this method is, to get values of higher accuracies, higher order factorial and power has to be calculated. Moreover to implement this we would at

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 2, Issue 4, July 2013

least require a multiplier, divider, adder and a subtractor. For good accuracy it would be required to take each term in calculation till they become insignificant. Thus this approach has a lot of hardware requirements as well as it is slow.

### *B. Look up Table*

The Lookup table approach involves storing values of sine and cosine at different angles. Based on the number of values stored, the lookup table can be big or small, but clearly, the smaller the lookup table, more is the error involved. The problem with a bigger lookup table is that it requires more memory and memory is expensive. Moreover the size of the Lookup table increases exponentially with the increase in the precision of the angle. Though this approach provides fast results it is very expensive to implement.

### *C. CORDIC*

CORDIC is an iterative algorithm capable of calculating trigonometric and various other functions. In this algorithm with the help of an adder/subtractor, a small look up table and a shifter the trigonometric functions can be calculated very easily. The advantage that CORDIC offers over other algorithms are that it does not require multiplication or division blocks, instead it works only a shifter, adder/subtractor and a small lookup table. This reduces the hardware requirement drastically and provides reasonably good speed. The main constraint in System on chip design is the amount of on chip memory and this constraint is equally valid in System on chip prototyping using programmable logic. Hence using CORDIC Algorithm we can minimize the memory requirements
Compared to other approaches, CORDIC is a clear winner when:

i. Hardware Multiplier is unavailable (e.g. microcontroller)
ii. To save the gates required to implement (e.g. FPGA)

## III. MODEL

The proposed architecture is shown in Figure 1 below.

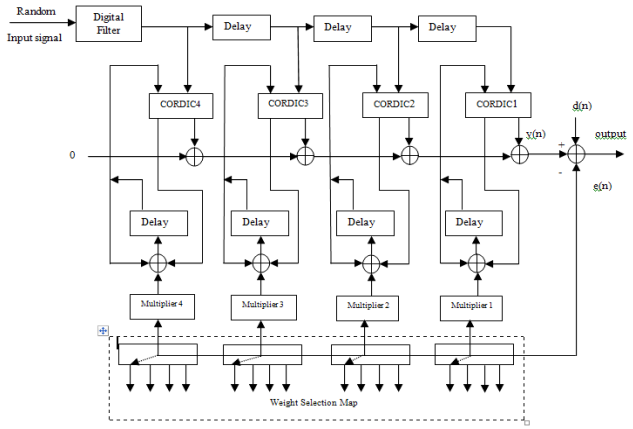


**Fig.1 Block diagram of Adaptive FIR filters with weight selection in various clutter medium**

The main requirement of Radar Receiver is to detect target amidst clutter environment. The clutter may be echo returns from stationary objects (Hills, electric poles etc.), vegetations or from other unwanted sources. During target detection, target echo return may accompany with clutters. In this paper, a bank of filters have been used in a required bandwidth of Doppler shift to detect target signal and all the filters are designed in such a way that clutters are cancelled adaptively. For circular rotation mode, CORDIC unit will give the output x sin θ and y cos θ for the signal inputs. The sinusoidal output term of CORDIC block is utilized for the purpose of filtering and other output of cosine term is used for the weight updation. In this architecture, pipelined multiplier is used. The performance of many computational problems is dominated by speed at which a multiplication is performed. The multiplier has two parts. One is for generating partial products and another for adding the partial products. With the pipelining technique, the propagation delay of the multiplier becomes the total delay of a single adder. That is why overall speed of signal processing is increased. The responses of the filters can be changed using a different set of weighting coefficients in Weight Selection Map. With reference to ground clutter, the following set of weights can be used:

a). Absence of clutter  Bank 0 ($W^0_1 W^0_2 \ldots\ldots W^0_N$)

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 2, Issue 4, July 2013

b). Weak clutter Bank 1 ($W^1_1$ $W^1_2$..... $W^1_N$)
c). Strong clutter Bank 2 ($W^2_1$ $W^2_2$..... $W^2_N$)
d). Very strong clutter Bank 3 ($W^3_1$ $W^3_2$..... $W^3_N$)

One reset clock is activated to initialize the system. Initially, the error at the output of final stage of filter is maximum. An automatic gate generator can be incorporated for the selection of weight bank as per the intensity of clutter. Once weight bank is selected, the error is fed back to the adaptive filter in association with the selected weights. The internal clock is adjusted in such a way that every sweep of signal is processed within single system clock. That means, the internal clock frequency is much higher than the system clock. In the filtering section, feedback error signal continuously updates incoming signals with previous reference signals and at the same time static returns or extremely slow moving target returns are also gets cancelled. Figure 1 shows the cancellation of noise which also is accompanied with target returns with reference to time index or number of iterations.

## IV. LMS ALGORITHM

Least mean squares (LMS) algorithms are a class of adaptive filter used to mimic a desired filter by finding the filter coefficients that relate to producing the least mean squares of the error signal (difference between the desired and the actual signal). It is a stochastic gradient descent method in that the filter is only adapted based on the error at the current time. It was invented in 1960 by Stanford University professor Bernard Widrow and his first Ph.D. student, Ted Hoff.

The least mean squares (LMS) algorithms adjust the filter coefficients to minimize the cost function. Compared to recursive least squares (RLS) algorithms the LMS algorithms do not involve any matrix operations. Therefore, the LMS algorithms require fewer computational resources and memory than the RLS algorithms. The implementation of the LMS algorithms also is less complicated than the RLS algorithms. However, the eigen value spread of the input correlation matrix, or the correlation matrix of the input signal, might affect the convergence speed of the resulting adaptive filter.

### *A. Adaptive transversal filters*

In a transversal filter of length N, as depicted in Figure 2 , at each time n the output sample y[n] is computed by a weighted sum of the current and delayed input samples x[n], x[n − 1], . . .

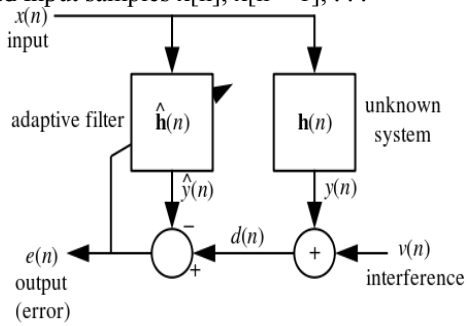


**Fig. 2 Adaptive transversal filters**

$$y[n] = \sum_{k=0}^{N-1} ck * [n]x[n-k] \quad (1)$$

Here, the $c_k[n]$ are time dependent filter coefficients (we use the complex conjugated coefficients $c_{k*}[n]$ so that the derivation of the adaption algorithm is valid for complex signals, too.

This equation re-written in vector form,

$$x(n)=[x[n],x[n-1],\ldots\ldots,x(n-N+1]]^T$$

the tap-input vector at time n, and

$$c[n] = [c_0[n] , c_1[n] ,\ldots\ldots\ldots\ldots, c_{N-1}[n]]^T$$

the coefficient vector at time n, is

$$y[n] = c^H[n]x[n] \quad (2)$$

Both x[n] and c[n] are column vectors of length N, $c^H[n]=(c^*)^T[n]$ is the hermitian of vector c[n] (each element is conjugated *, and the column vector is transposed T into a row vector).

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 2, Issue 4, July 2013

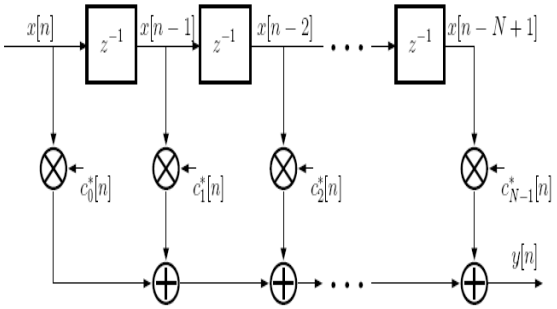


**Fig. 3 Transversal filter with time dependent coefficients**

In the special case of the coefficients c[n] not depending on time n: c[n] = c the transversal filter structure is an FIR1 filter of length N. Here, we will, however, focus on the case that the filter coefficients are variable, and are adapted by an adaptation algorithm.

### *B. Summary of the LMS algorithm*

1. Filter operation : $y[n] = c^H[n]\, x[n]$
2. Error calculation : $e[n] = d[n] - y[n]$
   where, d[n] is the desired output
3. Coefficient adaption : $c[n+1] = c[n] + \mu e^*[n]x[n]$

## V. Experimental Results And Performance Analysis

### *A. Simulation Results of Adaptive FIR Filter*

The test bench is developed in order to test the modeled design. This developed test bench will automatically force the inputs, which were taken from the reference, and will make the operations of algorithm to perform. The Figure 4 shows the simulation result for adaptive FIR filter for pulse doppler radar. Figure 4 shows the Simulink model of Adaptive FIR Filter for Pulse Doppler Radar.

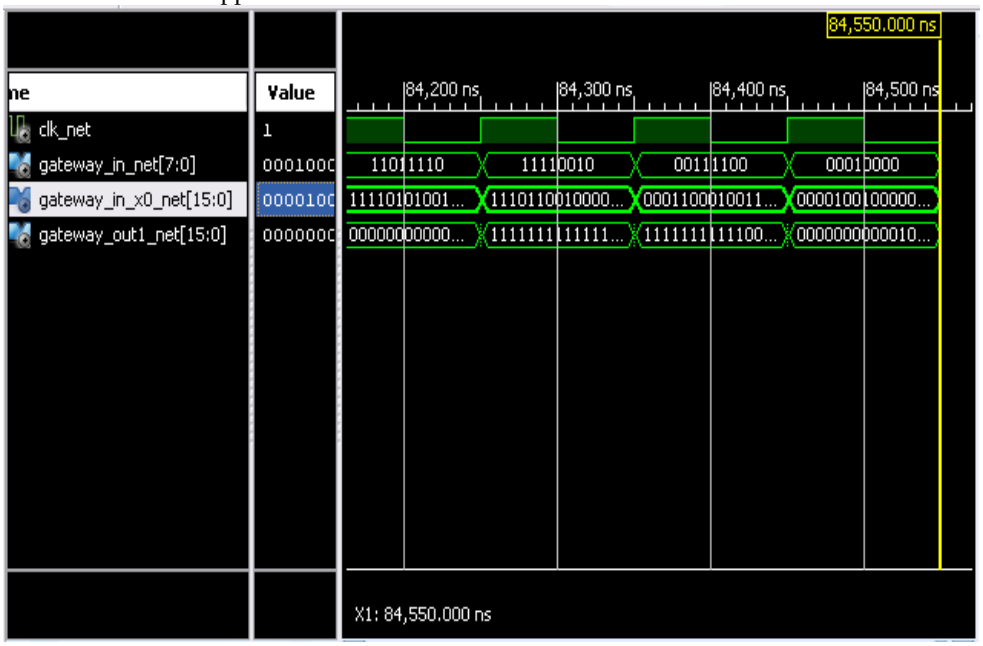


**Fig.4 Simulation result for adaptive FIR filter for pulse doppler radar**

It consists of four blocks: Input, digital filter, Adaptive filter. Here x(k) is the analog input signal which is applied to the digital filter (which is a cluttered signal) to convert analog to digital and the output of the digital filter block is "d(k)".

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 2, Issue 4, July 2013

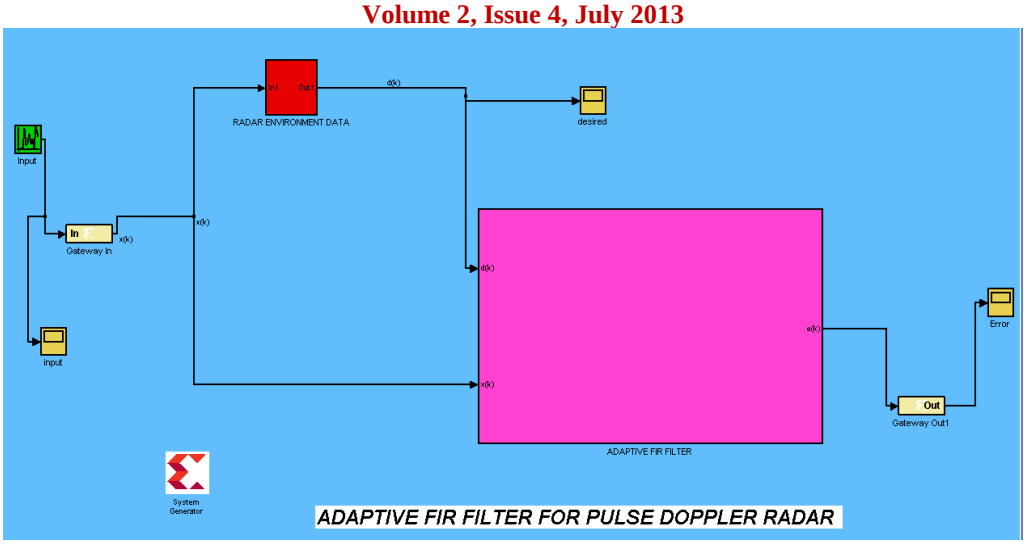


**Fig.5 Simulink model of adaptive FIR filter for pulse doppler radar**

### *B. RTL Schematic of Adaptive FIR Filter*

Apply x(k) the input random signal with 1024 samples and d(k) to the Adaptive FIR filter , As soon as we apply both inputs to the adaptive filter, then the adaptive filter starts comparing d(k) with x(k) and produces the output e(k). Figure 6 shows the RTL Schematic of Adaptive FIR filter where the arrangement of delay circuits, add sub structures, filters which helps to find out the errors efficiently. The adaptive filter can be realized for Doppler filtering function and thus play huge role in target detection and clutter cancellation. The filtering is achieved through a certain number of FIR filters. Within the radar receiver response curve, each FIR filter is centered on definite Doppler frequency, that is $f_d$ = 0 Hz to *PRF* where fd is the Doppler frequency and *PRF* is Pulse Repetition Frequency. The filters tuning on the desired Doppler frequency is achieved utilizing coefficients, called "weights". The 'weight' is complex in nature because the echo signal from target is also a complex signal. The number of coefficient needed is equal to number of filters. Each filter is tuned to a particular Doppler frequency. As a result, the Receiver frequency response curve is subdivided into a numbers of $f_d$ values.

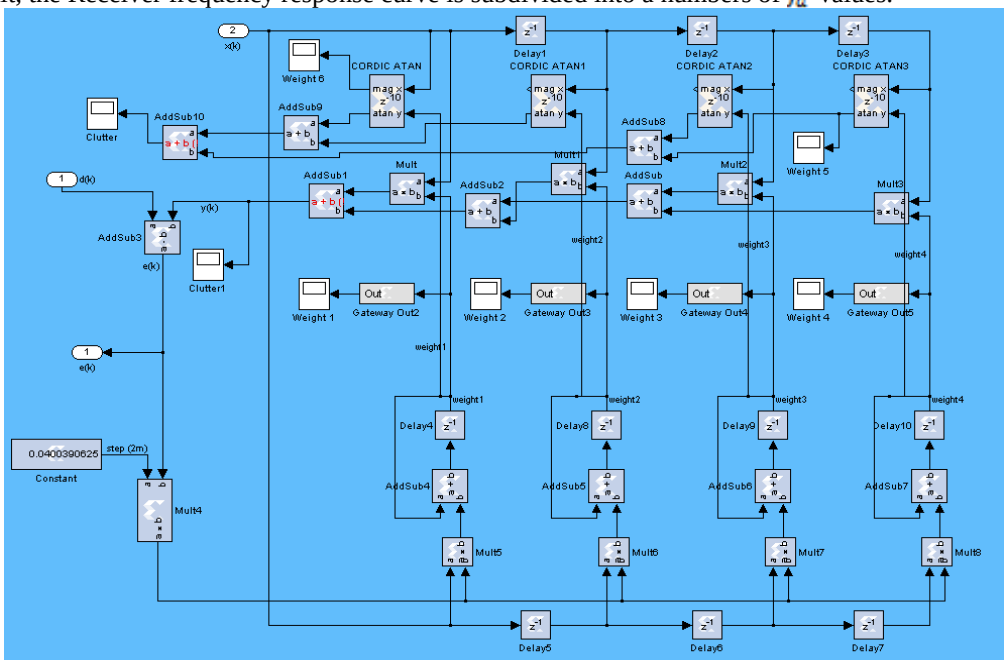


**Fig.6 RTL schematic of adaptive FIR filter**

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 2, Issue 4, July 2013

### *C. Simulation result of Adaptive FIR Filter in MAT-LAB*

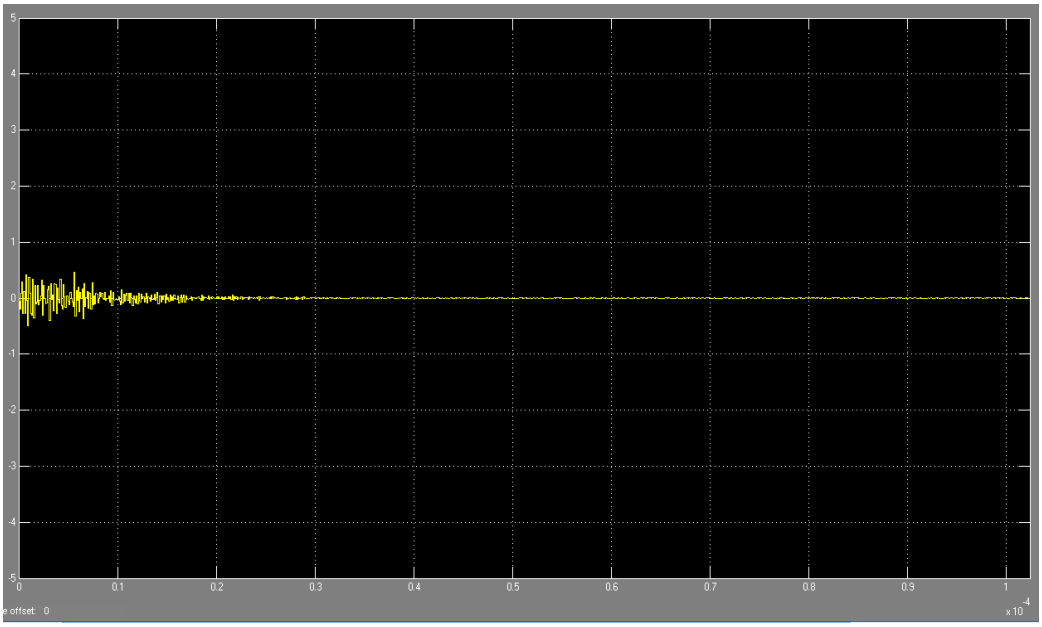

**Fig.7 Matlab output of adaptive FIR filter**

Initially error is more in the analog signal before it is processing or before applying to the Adaptive Filter. Thus when the signal is applied to the Adaptive FIR filter the error is reduced which is shown in Figure 7.

### *D. Analysis of Pipelined CORDIC*

The major components of the Pipelined CORDIC implementation consist of two carry look ahead adders, two multipliers, three groups of registers, and control logic. The control module is the only module that is coded behaviorally. The gate count and transistor count from the synthesis of the pipelined CORDIC implementation are shown in Table I. It is important to note that the timing information for the eight and sixteen bit is not given. There are two reasons that these numbers are left out. The first is that the eight and sixteen bit implementations do not require any arithmetic operations because they are implemented as pure function tables. For this reason, no code was written that could be statically timed. The second reason is that most local memory accesses can be made very quickly, even for large memories. Because the access time for the ROMs only has to be faster than the operating frequency of the algorithm, it can be made arbitrarily fast, especially for small tables.

**Table I: Pipelined CORDIC usage**

| Pipelined CORDIC | Bit Width | | | | | |
|---|---|---|---|---|---|---|
| | **8** | **16** | **24** | **32** | **48** | **64** |
| **Gate count** | N/A | N/A | 20,550 | 29,550 | 54,146 | 87,958 |
| **Trans. count** | N/A | N/A | 69,134 | 99,341 | 184,757 | 30,941 |
| **Density(%)** | N/A | N/A | 69 | 67 | 65 | 62 |
| **Delay(ns)** | N/A | N/A | 12.36 | 13.49 | 14.34 | 15.26 |
| **Latency(ns)** | N/A | N/A | 12.36 | 13.49 | 43.02 | 76.29 |

### *E. Simulation result of pipelined CORDIC*

Four CORDIC blocks are used in the FIR filter structure. As the weights get updated after every iteration, the error is reduced at output. The below Figure 8.a and Figure 8.b describes how the error is reduced after each CORDIC block.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 2, Issue 4, July 2013

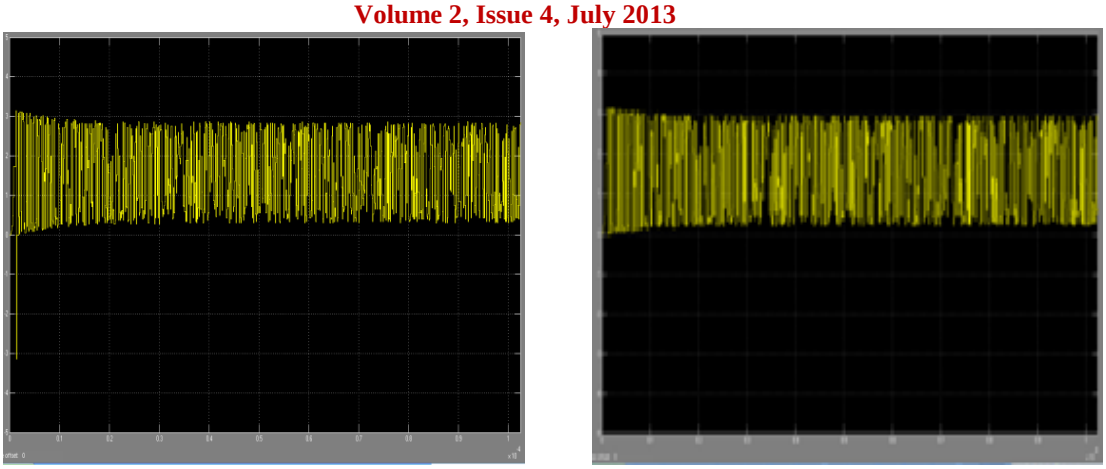

**Fig.8.a Output of CORDIC blocks**

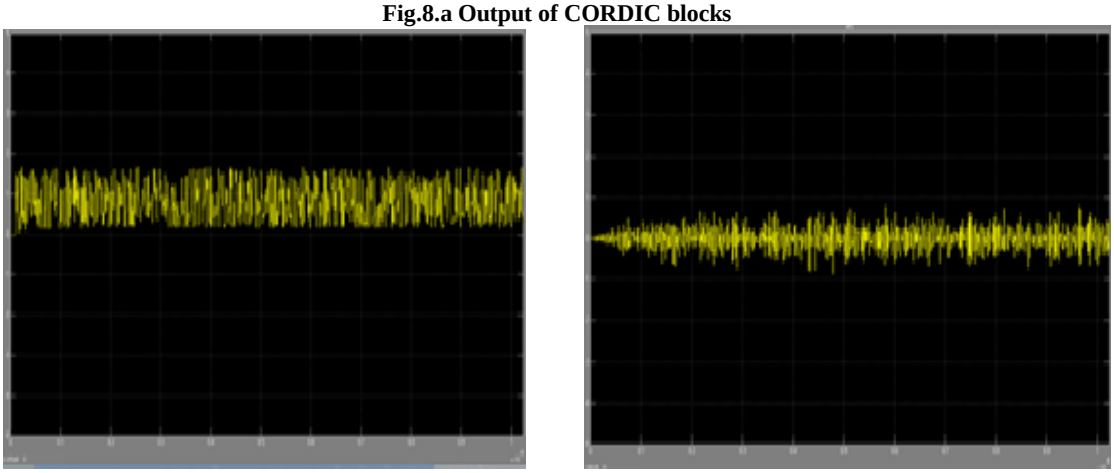

**Fig.8.b Output of CORDIC blocks**

***F. Performance Comparison with hybrid CORDIC***

The major components of the Hybrid implementation consist of twelve carry look ahead carry adders, twelve shifters, three groups of registers, and control logic. The only module that is coded behaviorally is the control module, which is a straightforward implementation of the unrolled CORDIC algorithm. The gate count and transistor count from the synthesis of the Hybrid CORDIC implementation are shown in Table II

**Table II: Hybrid CORDIC usage**

| Hybrid CORDIC | Bit Width | | | | | |
|---|---|---|---|---|---|---|
| | **8** | **16** | **24** | **32** | **48** | **64** |
| **Gate count** | 4,876 | 8,937 | 14,284 | 18,340 | 29,020 | 38,164 |
| **Transistor count** | 17,238 | 31,302 | 50,454 | 64,516 | 101,910 | 133,926 |
| **Density(%)** | 67 | 66 | 63 | 61 | 60 | 59 |
| **Delay(ns)** | 10.02 | 11.53 | 14.12 | 14.77 | 15.91 | 16.28 |
| **Latency(ns)** | 10.02 | 23.06 | 56.49 | 73.83 | 127.25 | 162.75 |

Not only the delay of the worst-case path reported, but the latency for obtaining the result is provided as well. The latency is determined by taking the worst-case path delay and multiplying it by the number of iterations that must be performed in order to calculate the results.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 2, Issue 4, July 2013

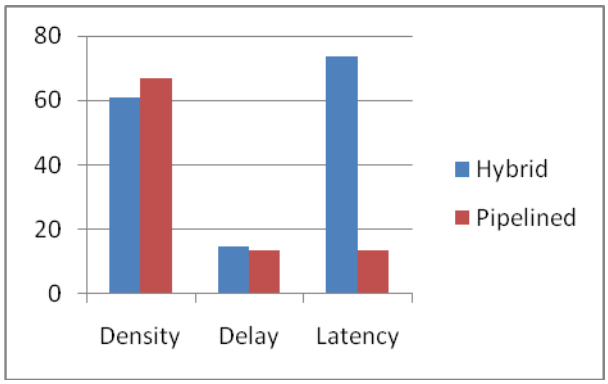


**Fig.9 Bar graph comparisons for 24 bit hybrid and pipelined CORDIC**

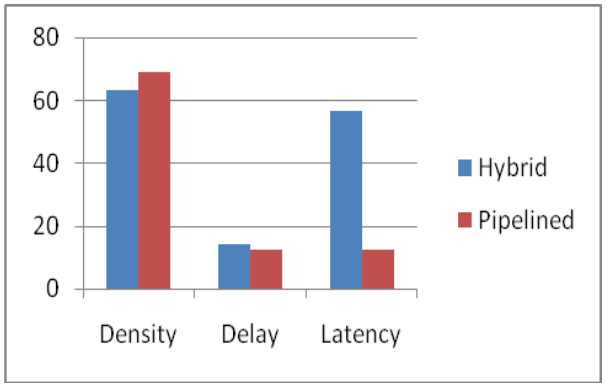


**Fig.10 Bar graph comparisons for 32 bit hybrid and pipelined CORDIC**

The Figure 9 and 10 clearly explains the comparison of hybrid and pipelined CORDIC. The density, delay and latency is less in Pipelined CORDIC when compared to Hybrid CORDIC and the percent improvement in the said parameters are tabulated in Table III

**Table III: Performance Comparison and Percent Improvement**

| Parameters | 24-bit | | | 32-bit | | |
|---|---|---|---|---|---|---|
| | Hybrid | Pipelined (proposed) | Percent Improvement | Hybrid | Pipelined (proposed) | Percent Improvement |
| **Density** | 63 | 69 | 10 | 61 | 67 | 10 |
| **Delay** | 14.12 | 12.36 | 07 | 14.77 | 13.49 | 07 |
| **Latency** | 56.49 | 12.36 | 77 | 73.83 | 13.49 | 81 |

## V. Conclusion And Scope For Future Work

### *A. Conclusion*

This project had presented architecture for detection of moving target amidst various intensity of clutter or noise using pipelined CORDIC unit in adaptive FIR filter banks. Through implementation and survey it has been found out that using pipelined CORDIC architecture as computational unit makes implementation of adaptive filters easier. Numbers of micro-rotations was adjusted to achieve speed of operation, while minimizing angle approximation error. The pipelined multiplier and CORDIC algorithm was used to achieve high throughput facilitating real time signal processing. The inherent issue of overflow in CORDIC structures was quite appropriately resolved using the proposed design.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 2, Issue 4, July 2013

From Table III it is clear that the pipelined based architectures take an edge over other structures and there is considerable improvement in the parameters like density, delay and latency. There was considerable improvement in area, density by a factor of 10, delay by a factor of 7, and latency by a factor of 77.

***B. Future scope***

Adaptive filtering algorithms are used for acoustic and noise cancellation and noise enhancer or clutter removal purpose. Selecting the adaptive filter that best meets our needs requires careful consideration. Two main considerations frame the decision:

1. Whether using an adaptive filter is a cost-competitive approach to solving our filtering needs?
2. It's suitability to filtering and signal processing application.

Three such areas are

- Filter consistency
- Filter performance
- DSP requirements

In future the parametric adaptive matched filter (PAMF) for space-time adaptive processing (STAP) is introduced via the matched filter (MF), multichannel linear prediction, and the multichannel LDU decomposition.

**ACKNOWLEDGMENT**

The joy and satisfaction that accompany the successful completion of any task would be incomplete without the mention of those who made it possible. I consider myself fortunate for having **Mr. Venkatesh Kumar N,** Asst. Professor, Dept. of ECE, Jain University, Bangalore as my internal guide who provided valuable support and guidance to make this project a success. Last, but not the least, I also take this opportunity to express my deep sense of gratitude to my parents and friends whose continuous support and invaluable suggestions has helped in the successful completion of my postgraduate program.

## REFERENCES

[1] N. Takagi, T. Asada and S. Yajima, "Redundant CORDIC Methods with a Constant Scale Factor for Sine and Cosine Computation", IEEE Trans. on Computers, vol. C-40, No. 9, pp. 989-995, 1991.

[2] Cochran D, "Algorithms and Accuracy in the HP35," Hewlett Packard Journal, vol. 23, pp. 10 – 11, 1972.

[3] Rheinstein John, "Simple Algorithms for Calculating Elementary Functions," IRE Transactions on Electronic Computing, vol. 8, pp. 142 – 145, August 1977.

[4] Cavallaro, Joseph R. and Franklin T Luk, "CORDIC Arithmetic for a SVD Processor," Proceedings of the 8th Symposium on Computer Arithmetic, vol.108, pp. 271 – 290, 1987.

[5] Linhai Xiang and Henk J Sips, "On-Line CORDIC Algorithms," IEEE Transactions on Computers, vol. 39, pp.1038 – 1052, 1990.

[6] Foley, James D, Andries van Dam, Steven K. Feiner, and John F. Hughes, "Computer Graphics – Principles and Practice," 2nd Edition, Addison Wesley, pp.367-897, 1995.

[7] J. E. Volder, "The Birth of CORDIC", International Journal on VLSI Signal Processing, vol.30, pp. 25-101, 2000.

[8] J. S. Walther, "The Story of Unified CORDIC", International Journal on VLSI Signal Processing, vol 25, pp.107-139, 2000.

[9] Henry Briggs and John Naphier, "Arithmetical Logarithmica", Excudebat Petrus Rammasenius, 1628.

[10] Andraka R.A., "Survey of CORDIC Algorithms for FPGA Based Computers", 6th International Symposium on FPGAs, pp 191-200, 1998.

[11] Jean-Claude Bajard, Sylvanus Kla, and Jean-Michel Muller, "BKM: A New Hardware Algorithm for Complex Elementary Functions", IEEE Transactions on computers, vol.43, no.8, August 1994.

[12] David J. Defatta, Joseph G. Lucas, and William S. Hodgkiss, "Digital Signal Processing: A System Design Approach", John Wiley and Sons, 1988.

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

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 2, Issue 4, July 2013

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

[15] S. Wang, V. Piuri and E. E. Swartz Lander. Jr. "Granularly-pipelined CORDIC processors for sine and cosine generators", IEEE International Conference on Acoustics, Speech ands Signal Processing, vol.6, pp. 3298- 3301, 1996.