**XXXXXXXXXXXXX**

XXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

**Abstract:** Co-ordinate Rotation digital computer may be a quick, simple, economical, and powerful algorithmic program used for various Digital Signal process applications. Primarily developed for real-time mobile computations, it uses a novel computing technique that is particularly appropriate for resolution the pure mathematics relationships concerned in plane co-ordinate rotation and conversion from rectangular to polar kind.We propose three reconfigurable CORDIC designs to compute spatial coordinates and orientation with high throughput: 1) a reconfigurable rotation-mode CORDIC that works either for circular or for hyperbolic trajectory; 2) a reconfigurable vectoring-mode CORDIC for calculating the circular and hyperbolic trajectories; and 3) a Generalized reconfigurable CORDIC that can work in any of the two modes for both circular and hyperbolic trajectories.

The reconfigurable CORDIC is designed in such a way which can helpcomputing the exponential as well as trigonometric functions, Also,square –roots,logarithms, and so on can be calculated using the circular and hyperbolic CORDIC by deploying it into one of the two mode i.e, either rotation-mode or vectoring-mode CORDIC in one single circuit. It can be used in scientific calculators, graphics processors, digital synchronizers and so on. It offers substantial saving of area complexity over the conventional design for reconfigurable applications.

***Index Terms*** -- Circular trigonometry, coordinate rotation digital computer (CORDIC), hyperbolic trigonometry, reconfigurable CORDIC.

**I. Introduction**

For a protracted time the sphere of Digital Signal process has been dominated by Microprocessors. this is often in the main as a result of the supply designers with the benefits of single cycle multiply-accumulate instruction further as special addressing modes. Although these processors square measure low-cost and versatile they're comparatively slow once it involves performing artscertain hard-to-please signal process tasks e.g. compression, electronic communicationand Video process. Of late, speedy advancements are created within the field of VLSI and IC style. As a result special purpose processors with custom-architectures have come back up.

Higher speeds may be achieved by these bespoke hardware solutions at competitive prices,to feature to the present, numerous easy and hardware-efficient algorithms exist that map well onto these chips and might be accustomed enhance speed and suppleness whereas performing arts the specifiedsignal process tasks. One such easy and hardware-efficient algorithmic rule is CORDIC, for Coordinate Rotation information processing system, projected by Jack E Volder. CORDIC uses solely Shift-and Add arithmetic with table Look-Up to implement completely different functions. By creating slight changes to the initial conditions and therefore the LUT values, it may be accustomed expeditiously implement pure mathematics, Hyperbolic, Exponential functions, Coordinate transformations etc. achieving a new profound hardware. Since it uses solely shift-add arithmetic, VLSI implementationof such Associate in Nursing algorithmic rule is possible. DCT algorithmic rule has various applications and is wide used for compression. Implementing DCT CORDIC algorithmic rule reduces the quantity of computations throughoutprocessing, will increase the accuracy of reconstruction of the image, and reduces the chip space of implementation of a processor engineered for this purpose. This reduces the powerconsumption.

FPGA provides the hardware atmosphere within which dedicated processors are often tested for his or her practicality. They perform varied high-speed operations that can't be completed by a straightforward microchip. theforemost uniqueness of FPGA is that it offers quick programmability. Because of which it is highly recommended forms the perfect platform to implement and check the practicality of anycumbersome processor designed using CORDIC rule.

Window filtering techniques are ordinarily used in signal process paradigm to limit time and frequency resolution. varied window functions are developed to suit completely different needs for side-lobe step-down, dynamic vary, then forth. Commonly, several hardware economical architectures are obtainable for realizing FFT, however a similar isn't true for windowing–architectures. the traditional hardware implementation of window functions uses operation tables that make to varied space and time complexities with increase in word lengths. Moreover, they are doing not enable user-defined variations within the window length. Associate in Nursing economical implementation of versatile and reconfigurable window functions using CORDIC rule is usually recommended. although they permit user-defined variations in window length, latency may be a major drawback. The CORDIC rule inherently suffers from latency problems and using 2 CORDIC processors nonparallel, as is done,the latency of the system is hampered.

**II. Literature Survey**

During the process of spectral analysis, the input signal needs to be truncated such that it implicates a finite observation window and suits the length of FFT processor. This direct truncation using standard windowing, referred to as rectangular window perform results in undesirable results referred to as spectral discharge and paling effect in frequency domain. To minimize these effects throughout spectral analysis, researchers have planned totally different styles of windowing functions like Hanning, playing and Blackman windowing functions. These windowing functions are wide adopted due to their sensible spectral characteristics like central peak breadth, 6-dB point, highest aspect lobe and rate of aspect lobe fall off and equivalent noise information measure (ENBW). Among these, Blackman windowing results in higher aspect lobe attenuation. it's gratuitous to gift of these characteristics intimately here, but readers could refer for an equivalent. Here solely Blackman windowing has been mentioned for implementation. although memory based mostly implementation is already existing, that restricts versatile implementation and additionally restricts fitting with the advanced FFT processors in terms of variable length and speed. Basic plan of this work is to propose a versatile and quick design for Blackman windowing perform to suit with the advanced FFT processor. Before presenting the planned design within the next section, Blackman windowing perform has been highlighted here in brief. A typical diagram for real time FFT based mostly spectral analysis system is shown in Fig.1.

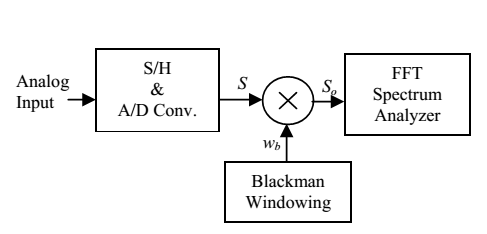


Fig.1. Spectral analysis system

The Blackman window, accompanied with the approximation coefficients, shows attenuation of a minimum of 60dB for aspect lobes[1] with solely a modest increase in computation over that needed by the Hanning and playing window due to a different cos term as in equation (2). This windowing operation demands the insight for planning hardware economical, versatile window length setting and high output VLSI design exploitation CORDIC whose implementation is kind of economic in terms of hardware. currently from equation (2), we have a tendency of having a parallel and pipelined design for foremost mentioned windowing operation, wherever the choice of window length (N) is user outlined as per demand for the appliance.

As the equation desires pure mathematics computation, therefore the implementation exploitation for CORDIC algorithmic rule is great selection in terms of computation and to vary the worth of N dynamically. In spite of it, search table or fixed storage methodology fails to realize identical, just in case of fastened N conjointly, the existing implementation is predicated on search table, it consumes longer to access the fixed storage and to cypher multiplication and addition. Whereas CORDIC based projected design mostly offers same result with high output and lesser hardware compared to fixed storage based computation. Here multiplication and pure mathematics computations are realised exploitation linear and circularCORDIC algorithmic rule severally

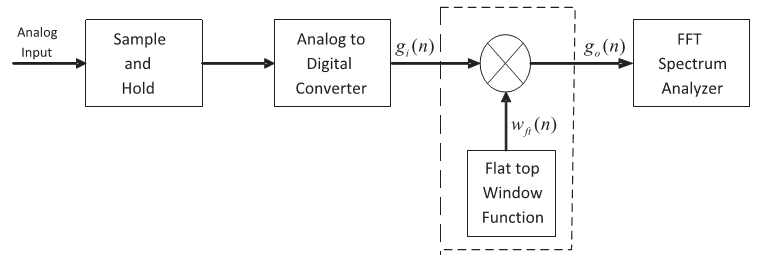


Fig.2. Block representation for spectral analysis

**III. Proposed System**

**RECONFIGURING CORDIC**

To design a reconfigurable CORDIC design with minimum reconfiguration overhead, we are required to maximize the sharing of common hardware circuit in numerous configurations. Therefore, to explore the likelihood of reconfigurable CORDIC, we examine, here, the commonalities in 3 main divisions for CORDIC implementation, namely: 1) the coordinate-rotation matrix; 2) choice of elementary angles; and 3) direction of small rotations.

**A. Reference Reconfigurable CORDIC**

A basic style for reconfigurable CORDIC supports a unique but a basic unified CORDIC algorithmic rule was planned. The key concern with the typical reconfigurable design is that the incompatibility in mythical monster of circular and hyperbolic trajectories. The mythical monster of circular CORDIC is [−99°, 99°], whereas that of hyperbolic CORDIC is given by |θ|≤1.1182 radians. This will limit the most angle of rotation of the reconfigurable design to 64°. The incompatible mythical monster of circular and hyperbolic CORDICs makes it troublesome to implement them within the same circuit to perform rotation through [−180°, 180°]. Another major issue with the traditional reconfigurable CORDIC is scaling. We want to possess 2 totally different scaling circuits for circular and hyperbolic CORDIC, and choose the output from one amongst the scaling circuits reckoning on the choice of mechanical phenomenon of operation.

**B. Design Stratergy incooperated for Proposed Reconfigurable CORDIC**

The circular and hyperbolic CORDICs need two totally different and unique scaling circuits, which can be sort of expensive. Therefore, it's would to great option to use a scale-free implementation within the reconfigurable CORDIC. Here, we have a tendency to discuss the scaling-free CORDIC and its limitations, followed by a detailed discussion on the style strategy for a reconfigurable CORDIC.

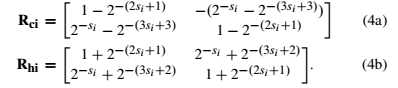
1. Scaling-Free CORDIC algorithmic program and Its Limitations: The scaling-free CORDIC [2] employs and uses second-order Taylor series approximation, wherever the rotation matrix is given by



This approximation imposes a restriction on the basic-shift of i =[(b−2.585/3)]. For 16-bit applications, the basic-shift is i =4, that reduces the rotation to 7.16°, which might be extended to 22.5° using multiple iterations resembling the basic-shift i =4. This is often a significant downside, that limits the pertinence of this rule. Moreover, the algorithms focus solely on circular rotation-mode, that can't be directly extended to hyperbolic CORDIC, since the second order of approximation of Taylor series enlargement of hyperbolic functions leads to a really low mythical creature (nearly 22.5°).But, thanks to the lack of symmetry in hyperbolic functions, the roc can't be extended to the complete coordinate house.

2) Reconfiguring the Rotation-Mode in CORDIC:

Scaling-free algorithms for circular and hyperbolic trajectories are planned. Moreover, in both the scaling-free algorithms, athird order approximation of Taylor series is employed to derive the CORDIC rotation-matrices, as



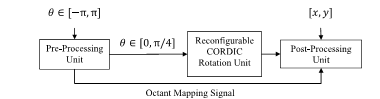


Fig.3. Proposed reconfigurable rotation-mode CORDIC processor.

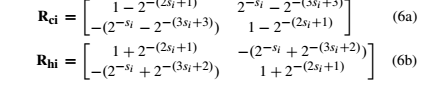
Note that a similar set of elementary angles is used for each circular and hyperbolic rotation-modes. this is often a big advantage in deriving the reconfigurable CORDIC, since no differentiation is needed to spot the small rotations in step with the trajectories. For circular and hyperbolic trajectories, the elementary angles are redefined as



Whereas i is the at the variety of shifts for the ith iteration. The mythical monster for each the trajectories is compatible and extends to the complete coordinate area. the look for rotation-mode CORDIC with slight modification is extended to support vectoring-mode as mentioned below

3) Reconfiguring the Vectoring-Mode of CORDIC:

To realize a vectoring-mode CORDIC into account, all the small rotations will be required to be performed within the clockwise direction for both the circular and hyperbolic trajectories. The rotation matrices are given by



Where i here is the shift-index of ith iteration . The sign-bit of the y-coordinate over successive iterations determines the angle of rotation θ. For vectoring-mode, the utmost angle of rotation that may be computed lies within the vary [0,π/4]. However, this variable can be extended to the whole coordinate area using measuring the instrumented wave symmetry of trigonometric function like the cos functions for circular flight.

**Proposed Reconfigurable CORDIC**:

The coordinate calculation matrices for circular and hyperbolic CORDICs dissent by the sign of operands, and to appreciate that additions are to get replaced by subtractions and vice-versa. this could be simply completed by a reconfigurable add/subtract circuit. In each cases, the basic-shift might be either two or three, but the amount of small rotations varies with the mode of operation. Besides, every case can have its own circuit to modify the extension of roc. based on these observations, we style 3 reconfigurable CORDIC architectures:

1) rotation-mode reconfigurable CORDIC;

2) vectoring-mode reconfigurable CORDIC;

3) generalized reconfigurable CORDIC.

**A**. **Rotation-Mode Reconfigurable CORDIC**

The projected vogue for reconfigurable rotation-mode CORDIC (shown in Fig. 3) consists of three parts: 1) preprocessing unit; 2) reconfigurable CORDIC rotation unit; and 3) post method unit. This foremost preprocessing unit ensures that the input rotation angle to the CORDIC method structure incessantly lies at intervals the vary [0,π/4], as a result of which the foremost rotation angle which can be handled by little rotation sequence generator is π/4. The post method unit is required only for circular trajectory to swap/complement the sine/cosine values betting on the instrument of the rotation angle. The user can also manage the trajectory of the reconfigurable CORDIC by driving a 1-bit signal T. The rotation matrix for reconfigurable rotation-mode CORDIC is obtained once unifying the rotation matrices of circular and hyperbolic case given by fig 4. as follows,



Where T= 0 hyperbolic

T=1 circular

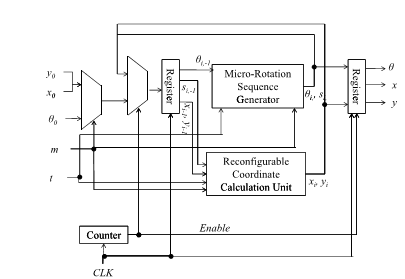


Fig.4.Block Structure of the proposed reconfigurable recursive CORDIC architectures.

1. **Proposed Recursive Architecture**:

The algorithmic design (shown in Fig. 4) uses only one CORDIC rotator unit to perform all the CORDIC iterations. The circular CORDIC also employs the need of use of one iteration but the hyperbolic CORDIC, however has the tendency to understand the design for an equivalent range of iterations (eight for sbasic=2 and eleven for sbasic=3) for each circular and hyperbolic trajectories. The reconfigurable coordinate calculation unit (RCCU) is shown in fig. 5;

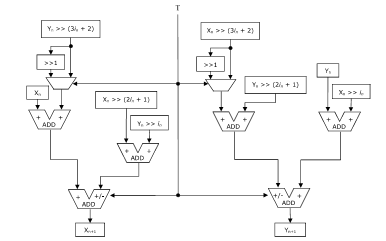


Fig.5. RCCU for recursive design

**2. Proposed Pipelined Architecture:**

Fig. 6 shows the reconfigurable CORDIC rotation unit for basic-shift . The shift-index siis mounted in each RCCU, and hence the shifters are hardwired and don't involve high quality barrel-shifters. The implementation of RCCUs varies with respect to the basic-shift si value applied. With slight modifications, the pipeline is extended for basic-shift three.

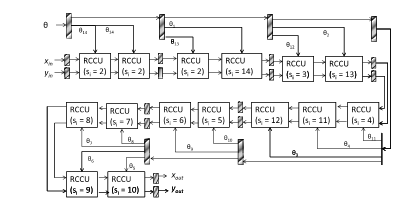


Fig.6. Reconfigurable rotation-mode CORDIC unit for basic-shift 2

**B. Reconfigurable Vectoring-Mode CORDIC**

The reconfigurable rotation matrix for vectoring mode is obtained by unifying (6a) and (6b), as



Where T=0 hyperbolic ----🡪8

T=1 circular

By ever-changing the implementation of the RCCU to implement, the recursive design of Fig. can be used to run CORDIC iterations for vectoring-mode. The change in counter to be made is 15 for sbasic=2, and 17 for sbasic=3. The pipelined design of vectoring-mode reconfigurable CORDIC consists of eight stages for sbasic = a pair of, as shown in Fig. 7.

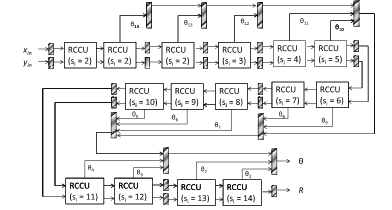


Fig 7 Proposed pipeline reconfigurable vectoring-mode CORDIC unit for sbasic=2

Similar to reconfigurable rotation-mode CORDIC, for increasing shift-indices, the implementation of RCCUs is simplified for reconfigurable vectoring-mode CORDIC further. The input coordinates [xin’,yin’] are initially preprocessed to get coordinates[xin,yin] and octant mapping signals. The coordinates [xin,yin] supplied as the input to the vectoring-mode CORDIC pipeline to get an angle θ ∈[0,π/4]. The rotation angle θ generated by the vectoring-mode CORDIC pipeline is mapped to the required octant using the octant mapping signals generated by the preprocessing unit.

Therefore, the roc supported by the projected vectoring-mode reconfigurable CORDIC is [−π, π].

**C. Projected Generalized Reconfigurable CORDIC :** Our new generic reconfigurable CORDIC unitcan be now deployed to work either in one among the two modes that is in vectoring-mode or in rotation-mode for each of the circular and hyperbolic trajectories. The user needs to provide the trajectory of operation by choosing a single bit signal T (T=1 for circular and T=0 for hyperbolic). Another single bit signal M is required to select the mode of either the rotational or vectoring mode of operation (M=0 for rotation-mode and M=1 for vectoring-mode). The algorithmic design of the projected generalized reconfigurable CORDIC is enforced by combining the CORDIC small rotators for each rotation-mode and vectoring-mode CORDICs, as shown in Fig. 8. The output of the projected algorithmic generalized reconfigurable CORDIC is that the same as that of the algorithmic reconfigurable vectoring-mode CORDIC. The block diagram for pipelined generalized reconfigurable CORDIC using basic-shifts basic=2 is shown in Fig. 9. It may be simply extended to basic-shifts basic=3 as is finished for reconfigurable rotation-mode and vectoring-mode CORDICs.

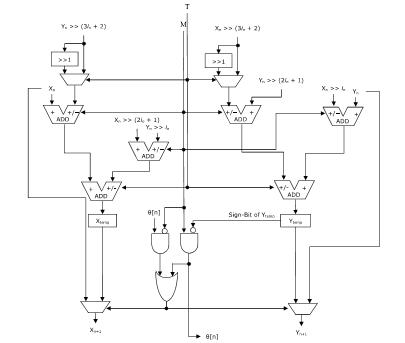


Fig.8. Structure of CORDIC micro rotator for the proposed recursive generalized reconfigurable CORDIC.

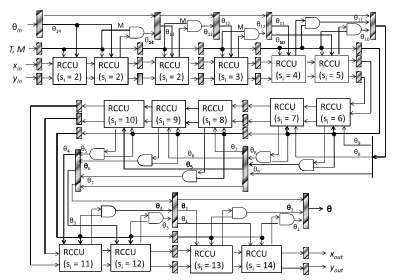
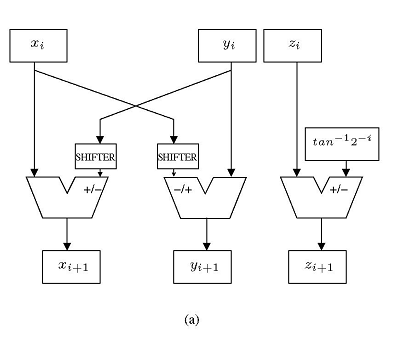
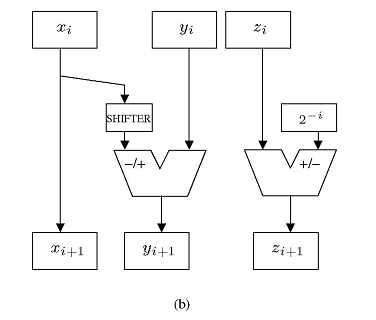


Fig.9. Proposed pipeline generalized reconfigurable CORDIC unit for sbasic=2.

**IV.Extension Work**

*Blackman window architecture*

**

**

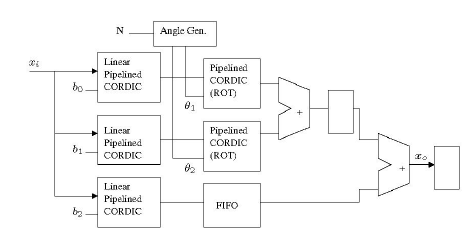
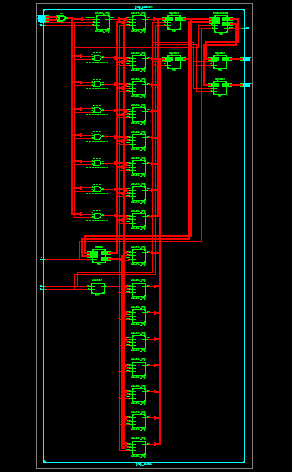


Fig. 10 Pipelined CORDIC Architecture (a) Circular (b) linear

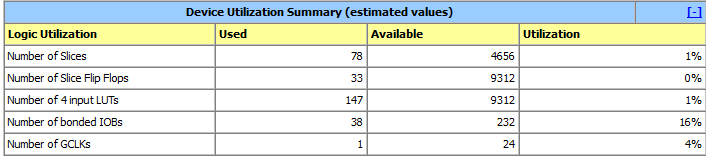
A VLSI architectural design for Black-man window is required, and most of the implementation of window functions for applications, can be solely achieved by the usage of DSP processors.Here the design is meant using major blocks like CORDIC(CO-ordinate Rotation DIgital Computer) and Han-Carlson adder. This design is unqiue in terms of window length. so one chip will be used for those applications, wherever variable length is needed.

The design is shown in Fig.10 for VLSI implementation of Blackman windowing perform. Major blocks of planned design area unit are described afterward. Two linear Coordinate rotation digital computer area unit which are used for multiplication of input samples with constant coefficientsthat is (b0 and b2), but the multiplication of constant coefficient (b1=0.5) with input samples is completed with solely onerous shifter (1-bit right) and passes through FIFO\_1 for synchronization with alternative parallel methods and equally FIFO\_2 is additionally used for synchronization. Circular CORDIC has been accustomed calculate cos functions given in equation (2) and multiplication of intermediate values (i.e. values from lower linear CORDIC and FIFO\_1 as shown in Fig.10). CORDIC blocks employed in our planned design area unit strictly pipelined, wherever add/sub circuit is that the vital path. Here length of FIFOs is adequate to the amount of stages of pipelined CORDIC minus one, e.g, for 16-bit preciseness CORDIC, the amount of stages area unit sixteen and therefore FIFO length to be fifteen.

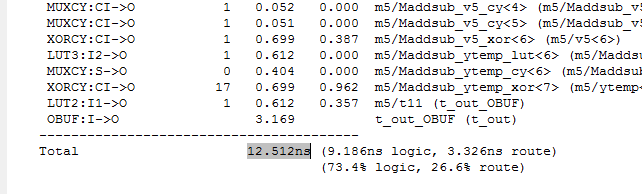
**V.RESULTS**



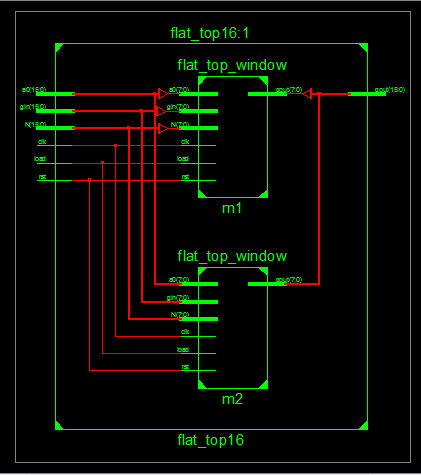
**Fig 11: Prop\_RTL**



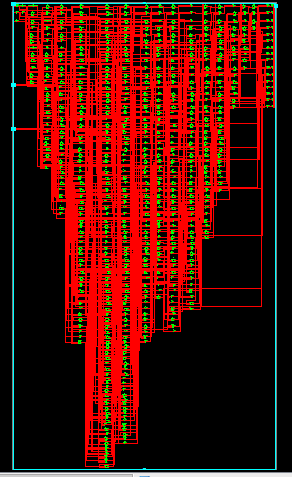
**Fig 12: Prop\_DUS**



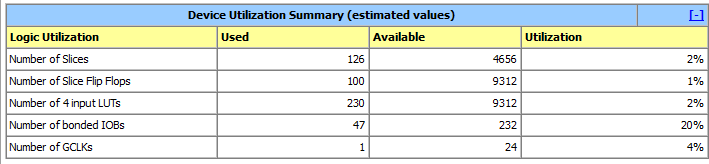
**Fig 13: Prop\_Delay**



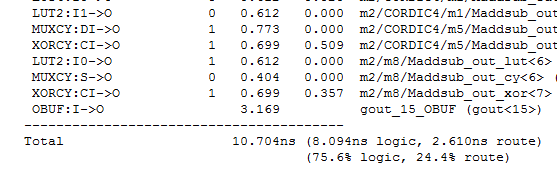
**Fig 14: EXT\_RTL**



**Fig 15: EXT\_Schematic**



**Fig 16: EXT\_DUS**



**Fig 17: EXT\_Delay**

**Conclusion**

CORDIC may be a powerful algorithmic rule, and a well-liked algorithmic rule of alternative once it involves various Digital Signal process applications. Computation of spatial co-ordinates and orientation using CORDIC-based processor such that high throughput is achieved provides us a robust mechanism of implementing advanced computations on a platform that gives lots of resources and adaptability at a comparatively lesser value.

Further, since the algorithmic rule is easy and economical the look and VLSI implementation of a CORDIC primarily based processor is well accomplishable. In this project a High through put CORDIC module for computing spatial coordinates and orientation algorithm is architected and simulated. Xilinx ISE is used as a synthesis tool. The output of the CORDIC core is analyzed and verified on the test-bench.

**References**[1]RayAndraka, FPGA '98. Proceedings of the 1998 ACM/SIGDA sixth international  
symposium on Field programmable gate arrays, Feb. 22-24, 1998, Monterey, CA. pp191-200  
[2]VikasSharma,"FPGA Implementation of EEAS CORDIC based Sine and Cosine  
Generator",M.Tech Thesis,Dept.Electron.Comm.Eng.,Thapar Uni., Patiala, 2009.  
[3]SatyasenPanda,"Performance Analysis and Design of a Discrete Cosine Transform  
Processor using CORDIC Algorithm",M.TechThesis,Dept. Electron.Comm.Eng.,NIT  
Rourkela, Rourkela, Orissa,2010.  
[4]Kris Raj, GauravDoshi, Hiren Shah [online] Available:  
http://teal.gmu.edu/courses/ECE645/projects\_S06/talks/CORDIC.pdf(URL)  
[5] S.K.Pattanaik and K.K.Mahapatra,'DHT Based JPEG Image Compression Using a Novel  
Energy Quantization Method", IEEE International Conference on IndustrialTechnology,pp.2827-2832,Dec 2006.

[6] Volder, J.,"Binary Computation Algorithms for coordinate rotation and function  
generation,"Convair Report IAR-1 148 Aeroelectrics Group, June 1956.  
[7] Volder, J.,"The CORDIC Trigonometric Computing Technique,"IRETrans.Electronic  
Computing, Vol EC-8, pp330-334 Sept 1959.  
[8] Walther J.S.,"A unified algorithm for elementary functions," Spring Joint Computer  
Conf.,1971,proc.,pp.379-385.