International Journal of Engineering Science Invention (IJESI)
ISSN (Online): 2319 – 6734, ISSN (Print): 2319 – 6726
www.ijesi.org ||Volume 7 Issue 5 Ver VII || May 2018 || PP 47-55

High Speed Algorithm for Arithmetic Operations Performed For Feature Extraction in Activity Recognition

SupriyaKurlekar
ShitalGhorpade,MadhuriSachane(SITCOE,Yadrav)

Abstract:*Human activity recognition is one of the most promising research topics for a variety of areas, including pervasive and mobile computing [1] [2], surveillance-based security [3], context-aware computing and ambient assistive living. It has, recently, received growing attention attributing to the intensive thrusts from the latest technology development and application demands. Over the past decade, sensor technologies, especially low-power, low-cost, high-capacity, and miniaturized sensors, wired and wireless communication networks and data processing techniques have made substantial progress. The advances and maturity of these supporting technologies have pushed the research focuses of the aforementioned areas to shift from low-level data collection and transmission toward high-level information integration, context processing, and activity recognition and inference. At the same time, solutions for a number of real-world problems have become increasingly reliant on activity recognition.Through this paper we have presented cordic based basic arithmetic operations required for feature extraction used as main block in Activity recognition.*

Date of Submission: 18-05-2018 Date of acceptance: 04-06-2018

I. Introduction

Activity recognition has gained prominence in the research community in wellbeing andhealthcare as well as sports training and monitoring applications.The main objective is to develop low power low complexity frame work for feature extraction and classification for activity recognition using Hardware efficient Algorithm.Through this paper we have presented cordic based basic arithmetic operations required for feature extraction.

II. Literature survey

Sensor network has revolutioned the modern world [2].various types of sensors such as accelerometers,gyroscope,magnetometers have been used for AR. Another approach being used is fusion data from vision systemand inertial sensors. it is mainly restricted to indoor activities under the unhindered surveillance of the vision system. Inclinometers and goniometers are other types of sensorsthat are used to measure upper/lower limbs kinematics.[3] even though there are potential gains of the remote monitoring system using wearable sensors , wearable size and energy consumption is big challenge need to be addressed.the number of sensors and their placement on human body have direct impact on AR. Recently vision based systems have used kinect sensor(by Microsoft) with depth finding camera. Inability to record moving object at long distance along with shadowing of the points of interest are drawbacks found[4]. Three major steps in activity recognition are**Sensorplacement,Data preprocessing ,Data classification** .

Conventionally supervised classification technique(K-nearest neighbor,support vector machine(SVM),Gaussian mixture model,random forest) and unsupervised technique namely k-means,GMM,Hiddenmarkov model(HMM) are used for classification.

Data preprocessing step mainly includes feature extraction . This is done in two ways time domain and frequency domain .Typical features required for the major classification problems are mean,rms,std. deviation, index of dispersion,kurtosis,skewness,absolutediff.,information entropy.[2] These features are extracted from long data series using computationally intensive statistical technique. Since energy consumption is proportional to the arithmetic complexity implementation of such processes in Sensor network node consumes significant amount of energy affecting the battery life[3]. Therefore it is of paramount importance to develop a low power strategy for feature extraction and classification in the resource constrained environment of sensor network for activity recognition.[4]

III. Prposed work

Computationally intensive steps are Feature Extraction (from data acquired by sensors),Data classification To be performed at sensor node itself for compensating significant energy required at the radio frontend of the sensor for continuous data transmission.

Energy consumption is proportional to the arithmetic complexity SN consumes significant amount of energy affecting battery life is major roadblockTo develop low power low complexity frame work for feature extraction and classification for activity recognition using Hardware efficient AlgorithmThere is need of robust energy efficient technique to extract motion features and processing strategies. Most of the features require the basic arithmetic operations like addition,subtraction,multiplication,division,square root and logarithm for successful computation. With the recent advances in VLSI,several effective low power design techniques have been proposed including hardware efficient algorithms for various operations. These algorithms provide a good trade off between accuracy and hardware complexity addressing speed and low power expectionsdemandes by signal processing applications.

CORDIC Algorithm can be used for performing basic operations like square root ,multiplication,division,addition,subtraction required to be performed in feature extraction process .Following implementation of cordic based calculations needed for feature extraction are obtained . CORDIC in circular plane and linear plane is designed and verified in the form of pipelined architecture for this purpose.

3.1vectoring Mode Of Cordic Algorithm

In vectoring mode the CORDIC rotator rotates the input vector through whatever angle is necessary to align the result vector with the X-axis.

The result of the vectoring operation is a rotation angle and the scaled magnitude of the original vector (the x component of the result).

The vectoring function works by seeking to minimize the y component of the residual vector of each rotation. The sign of the residual y component is used to determine which direction to rotate. Next if the angle accumulator is initialized with zero. It will contain the transverse angle at the end of the iterations.

In vectoring mode the equations are

$$\mathbf{x_{i+1} = x_i - y_i d_i 2^{-i}}$$

$$\mathbf{y_{i+1} = y_i + x_i d_i 2^{-i}}$$

$$\mathbf{z_{i+1} = z_i - d_i \tan^{-1}(2^{-i})}$$

Where d_i = +1 if y_i<0 , -1 otherwise.

Then

$$\mathbf{x_n = A_n \sqrt{x_0^2 + y_0^2}}$$

$$\mathbf{y = 0}$$

$$\mathbf{z_n = z_0 + \tan^{-1}(y_0/x_0)}$$

$$\mathbf{A_0 = \prod_n \sqrt{1 + 2^{-2i}}}$$

3.1.1 Arctangent

The arctangent θ = Atan(y/x) is directly computed using the vectoring mode CORDIC rotator if the angle accumulator is initialized with zero. The argument must be provided as ratio expressed as a vector (x,y). Presenting argument as a ratio has the advantage as being able to represent infinity (by setting x=0). Since the arctangent result is taken from angle accumulator the CORDIC rotator growth does not affect the result.

$z_n = z_0 + \tan^{-1}(y_0/x_0)$

3.1.2 Vector Magnitude

The vectoring mode CORDIC rotator produces the magnitude of the input vector of a by product of computing the arctangent. After the vectoring mode rotation the vector the vector is aligned with the x-axis. The magnitude of the vector is therefore the same as the x component of the rotated vector.

$$x_n = A_n \sqrt{x_0^2 + y_0^2}$$

The result is scaled by the processor gain which needs to be accounted for elsewhere in the system. This implementation of vector magnitude has a hardware complexity of roughly one multiplier of the same width.

3.2 VHDL implementation

CORDIC pre- processor is developed .for this mode x and y are the i/ps. In pre_ processing absolute value of x and y is determined .the sign of x and y is stored as status information which isequired by post processor for positioning the resultant in correct quadrant.

Main CORDIC block is same as developed for rotation mode only condition for micro rotation is based on 'sign of y'.

In CORDIC post_processor gain adjustment is done. The resultant radious is scaled by gain this is compensated by dividing by the same factor. In this correct quadrant depending upon sign of x and y is also adjusted.

For this entire rectangular to polar conversion task five modules proc.vhd, pre.vhd, CORDIC.vhd, CORDICpipe.vhd, post.vhd are developed.

The basic hardware remains same for CORDIC block. The pipelined structure is also same. Radious scaling and angle representation in appropriate quadrant is done.

Entity description

The basic hardware remains same for CORDIC block. The pipelined structure is also same. Radius scaling and angle representation in appropriate quadrant is done.

Main entity r to p

The main entity is shown below for vectoring mode. As shown in fig input x and y is given. Clk and ena is applied. About gives atan(y/x and Rout gives sqrt(x^2+y^2).

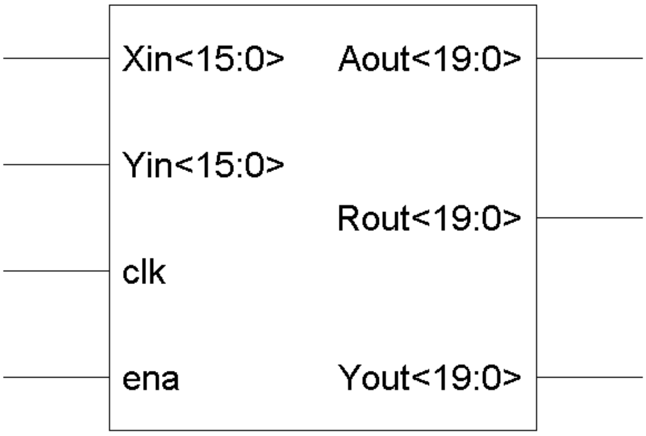


Fig 5.1 main entity vectoring mode

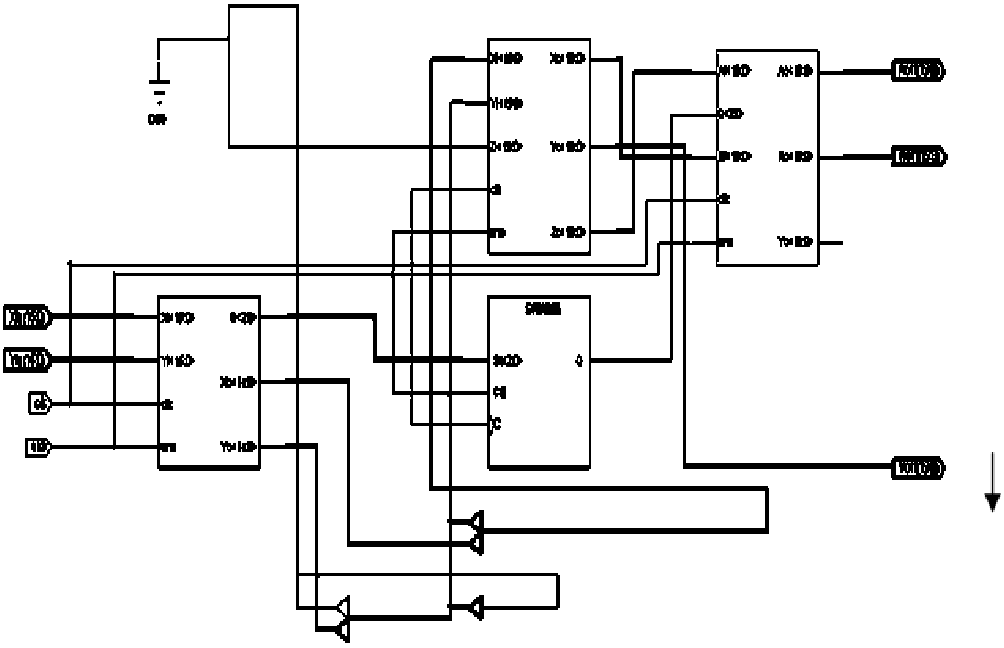


Fig 5.2 RTL main

3.2 Pre_Processing Block

As shown in fig Xi and Yi i/ps are given from main entity. Pre_Processing block will give absolute value of x and y in Xo and Yo ,Q vector stores status of sign bit of x and y also it keeps information about swapping of x and y.

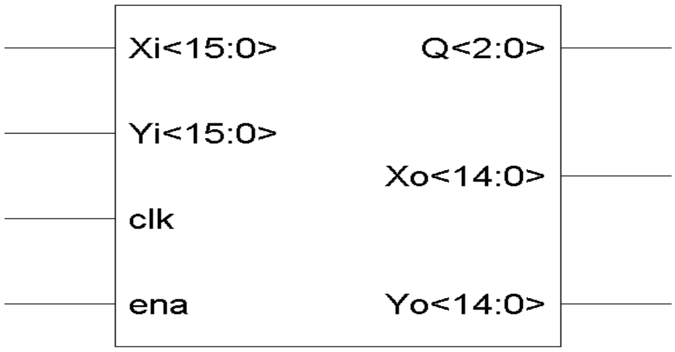


Fig 3.3 Entity Pre_processing

PRE PROCESSING

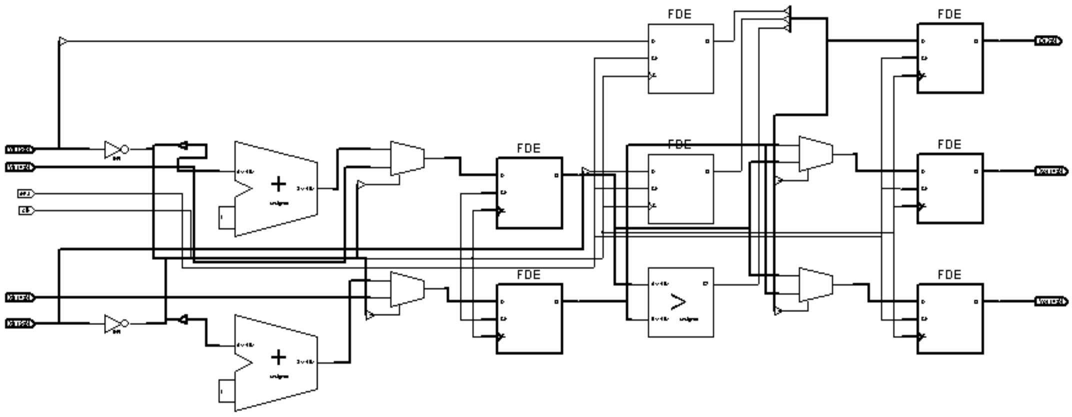
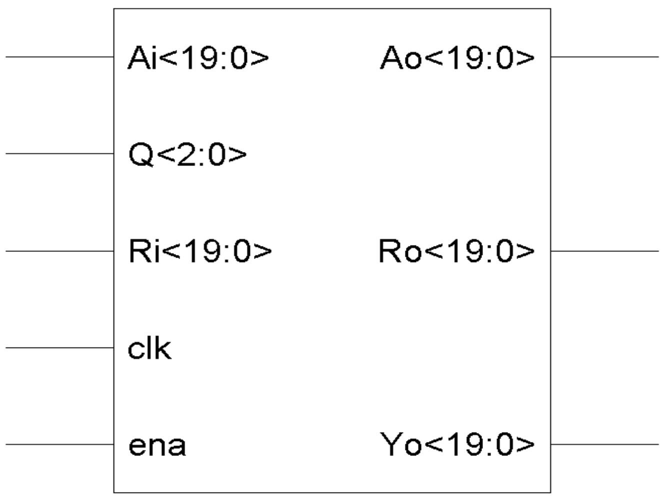


Fig 3.4 RTL Pre

3.4.3 Post_ Processing Block

Post entity accepts 'Ai' and 'Ri' from CORDIC final pipe calculated by CORDIC algorithm. After adjusting gain and quadrant it gives o/p in Ao and Ro.



5.5 Simulation Result

I/P Xin=3 =>"0000000000000011" Yin= 4=>"0000000000000100"
Arctan(y/x)=53.1301 Radious=sqrt(x^2+y^2)=5.0
O/P Rout =>"0000000000000101 . 0101" upper 16 bits integer part and lower 4 bits represent fractional part Rout=5.3125
Aout =>"00100101101111110011"
actual angle=[dec(Aout)/2^20]*360=arctan (y/x)=53.0814

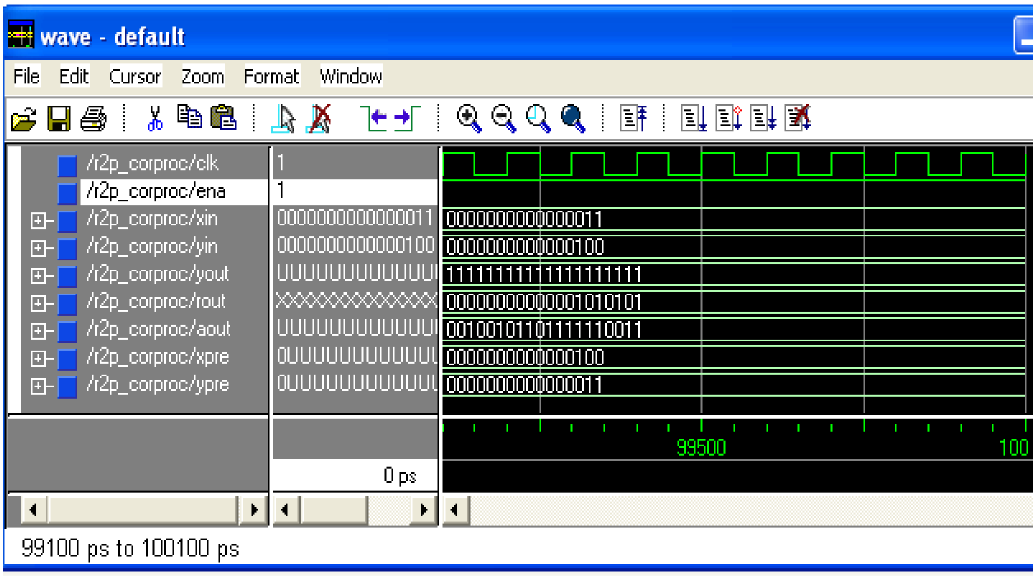


Fig. 5.7 Simulation for I/P result(x,y)=>(3,4)

I/P Xin= 3 =>"0000000000000011" Yin=- 4 =>"1111111111111100"
Arctan(y/x)=-53.1301 Radious=sqrt(x^2+y^2)=5.0

O/P Rout =>"0000000000000101 . 0101" upper 16 bits integer part and lower 4 bits represents fractional part Rout=5.3125
Aout =>"11011010010000001101"
actualangle=[dec(Aout)/2^20]*360=arctan(y/x)=306.9185=>
(360-306.9185)=>53.0814

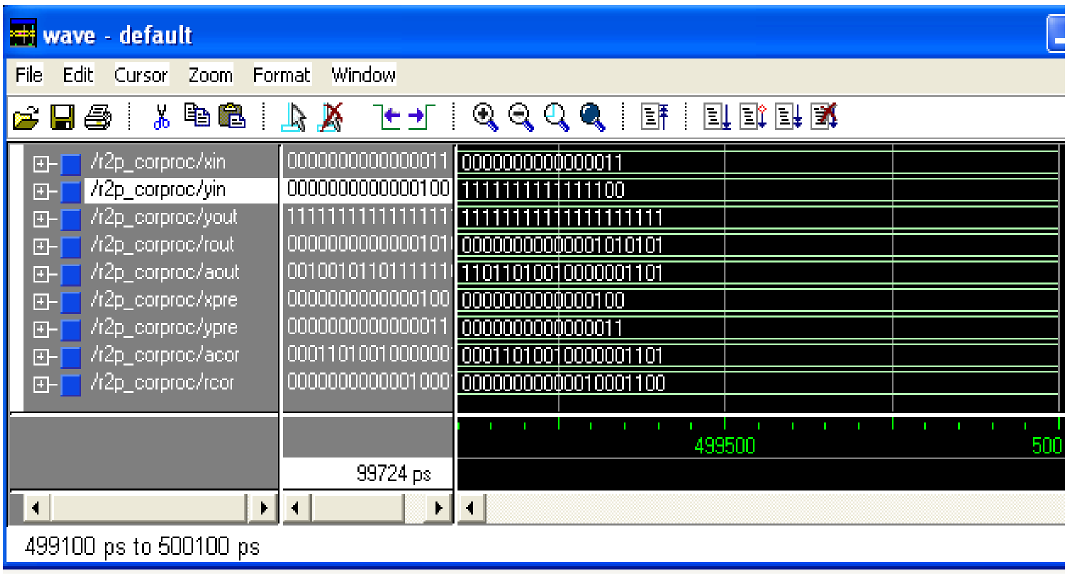


Fig. 5.8 Simulation for I/P (3,-4)

I/P Xin= -3 =>"1111111111111101" Yin= 4 =>"0000000000000100"
Arctan(y/x)=-53.1301 Radious=sqrt(x^2+y^2)=5.0

O/P Rout =>"0000000000000101 . 0101" upper 16 bits integer part and lower 4 bits represent fractional part
Rout=5.3125
Aout =>"01011010010000001101"
actualangle=[dec(Aout)/2^20]*360=arctan(y/x)=126.9185=>(180-26.9185)=>53.0814

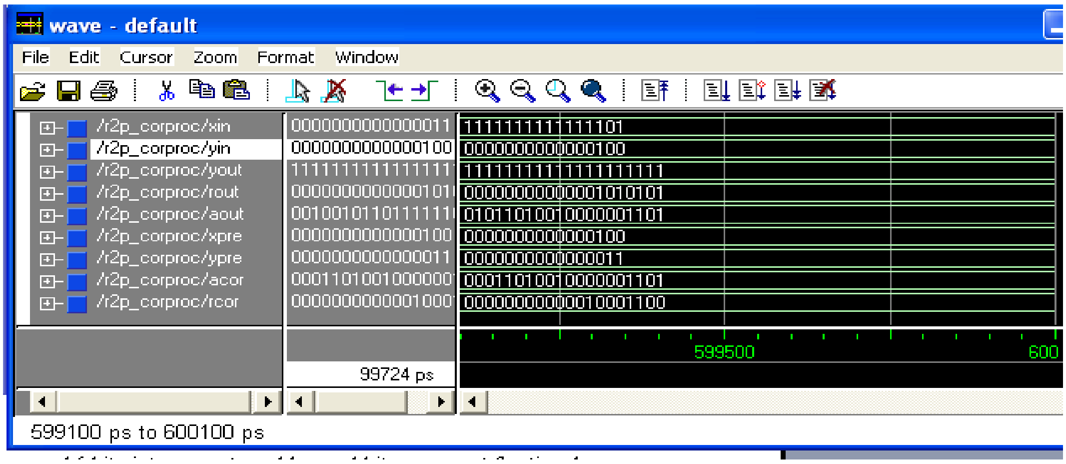


Fig. 5.9 Simulation for I/P result(x,y)=>(-3,4)

I/P Xin= -3 =>"1111111111111101" Yin= -4 =>"1111111111111100"
Arctan(y/x)=53.1301 Radious=sqrt(x^2+y^2)=5.0

O/P Rout =>"0000000000000101 . 0101" upper 16 bits integer part and lower 4 bits represent fractional part
Rout=5.3125
Aout =>"10100101101111110011"actual angle=[dec(Aout)/2^20]*360=arctan (y/x)=233.0814

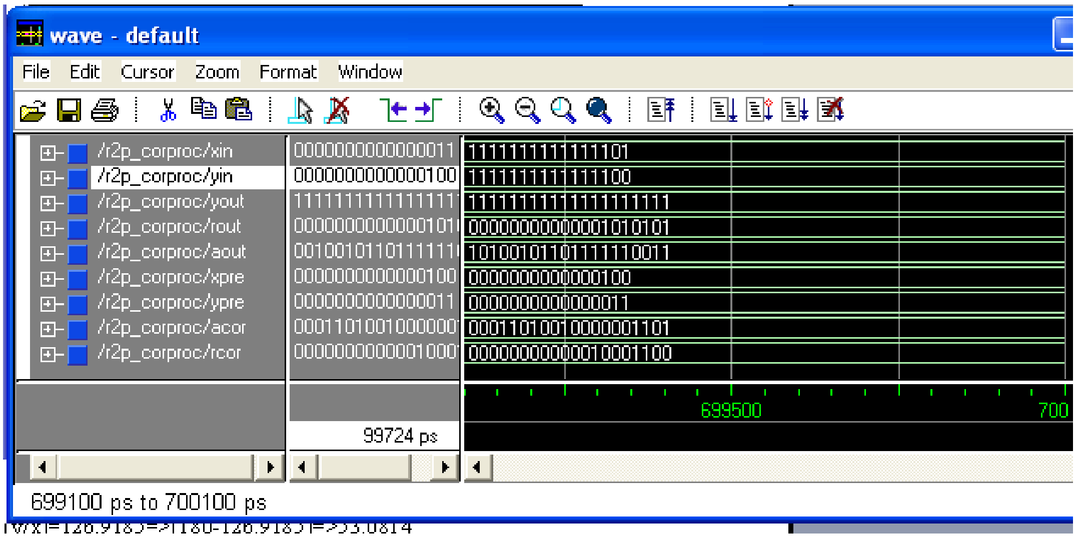


Fig .3.10 Simulation for I/P result(x,y)=>(-3,-4)

5.5.1 Synthesis Result
Device utilization summary:selected Device : 3s400pq208-5

Number of Slices:	639	out of	3584	17%
Number of Slice Flip Flops:	1108	out of	7168	15%
Number of 4 input LUTs:	1181	out of	7168	16%
Number of bonded IOBs:	93	out of	141	65%
Number of GCLKs:	1	out of	8	12%

==

Timing Summary(3s400pq208) Speed Grade: -5
Minimum period: 5.322ns (Maximum Frequency: 187.899MHz)
Minimum input arrival time before clock: 5.500ns
Maximum output required time after clock: 5.106ns
Maximum combinational path delay: No path found

Device utilization summary: selected Device : 3s50pq208-5

Number of Slices:	639	out of	768	83%
Number of Slice Flip Flops:	1108	out of	1536	72%
Number of 4 input LUTs:	1181	out of	1536	76%
Number of bonded IOBs:	93	out of	124	75%
Number of GCLKs:	1	out of	8	12%

Timing Summary(3s50pq208-5) Speed Grade: -5
Minimum period: 5.322ns (Maximum Frequency: 187.899MHz)
Minimum input arrival time before clock: 5.500ns
Maximum output required time after clock: 5.106ns
Maximum combinational path delay: No path found

Extension to Linear function(Rotation mode)
A simple modification to the CORDIC equations permits a computation of linear Functions.
$X_{i+1} = X_i - 0 . Y .d_i. (2^{-i}) = X_i,$ $Y_{i+1} = Y_i + X_i . d_i. (2^{-i})$
$Z_{i+1} = Z_i - d_i .(2^{-i})$

By rotation method,where d_i = -1, if $Z_i < 0$, +1 otherwise
which produces the result $X_n = X_0, Y_n = Y_0 + X_0. Z_0$, $Z_n = 0$
5.6.1 Simulation result
I/P Xin = “000011.0000000000”=> “3.0”
Yin = “000100.0000000000”=> “4.0”
Zin = “000000.1000000000”=> “0.5”
O/P Xo = “3.0” Yo = “5.5”

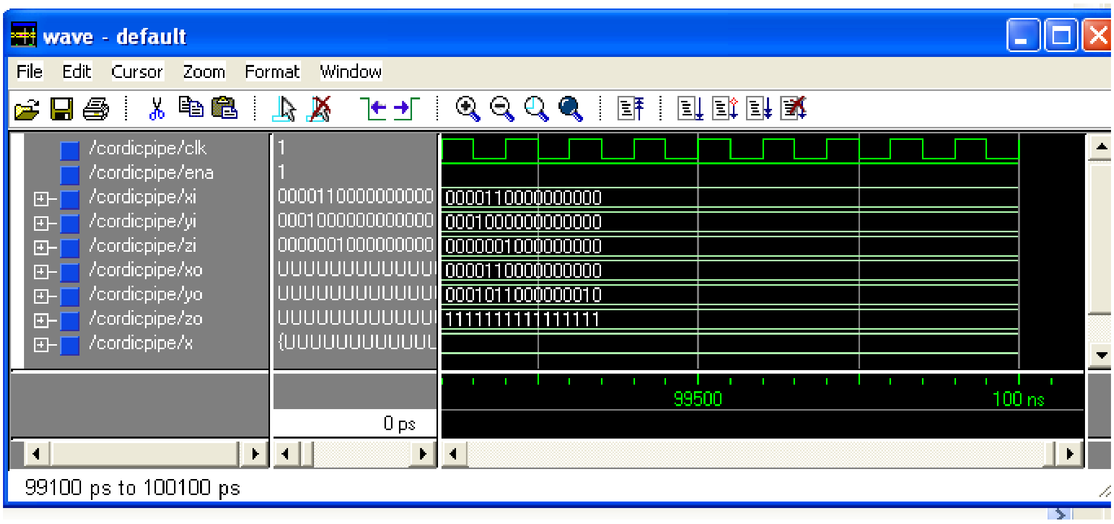


Fig.3.11 Linear plane(Rotation mode)

Extension to Linear function(Vectoring mode)
A simple modification to the CORDIC equations permits a computation of linear Functions.

$$x_{i+1} = x_i - 0.y_i.d_i.2^{-i} = x_i$$

$$y_{i+1} = y_i + x_i..d_i.2^{-i}$$

$$z_{i+1} = z_i - d_i.(2^{-i})$$

The vectoring mode (di=+1 if yi<0, -1 otherwise)is more interesting, as it provides a method for evaluating ratios.

$$\mathbf{x_n = x_o}$$

$$\mathbf{y_n = 0}$$

$$\mathbf{z_n = z_o - y_0 / x_0}$$

The rotations in the linear **co**-ordinate systems have a unity gain so no scaling corrections are required.

5.7.1 RTL Vectoring_Linear

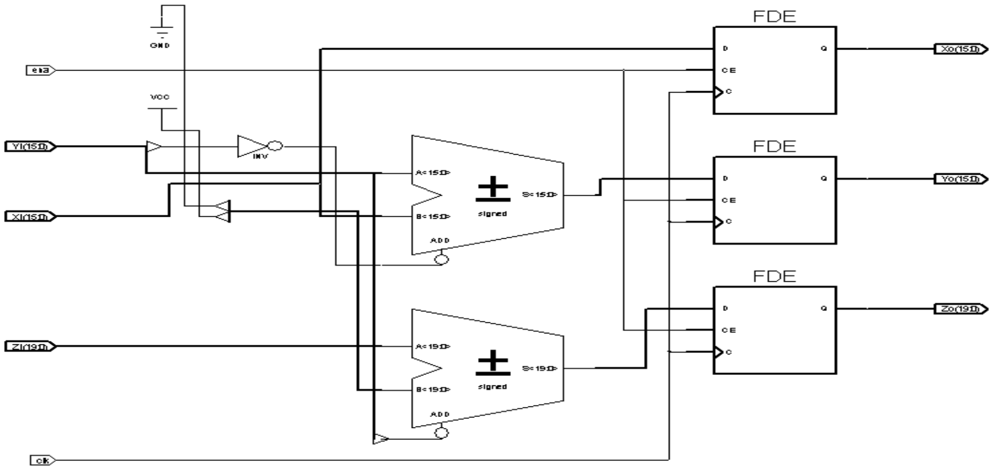


Fig 5.12 RTL Vectoring_linear

3.2 Simulation Result

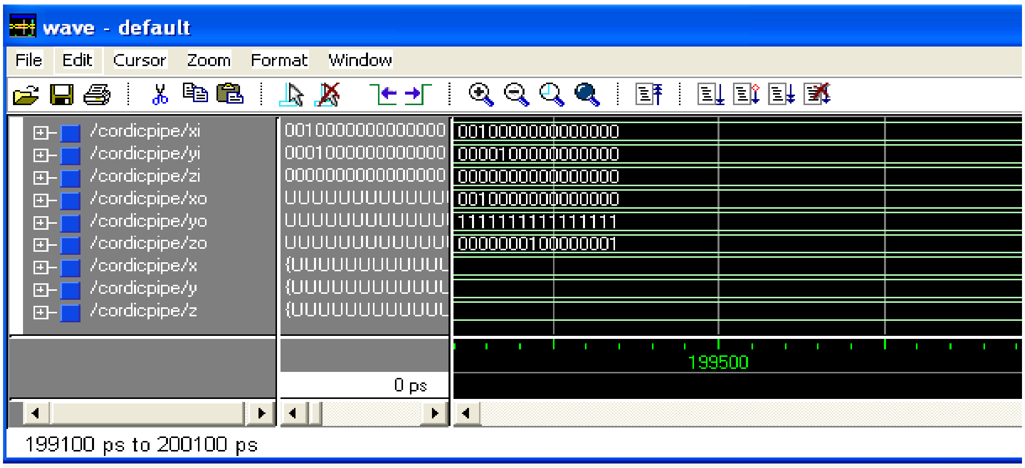


Fig 3.13 Result(2/8)=0.25

IV. Conclusion-

Cordic can be efficiently used as one building block for feature extraction calculation part as hardware efficient engine in Activity recognition applications

Referances

[1]. Dwaipayan Biswas, KoushikMaharatna, Member, **IEEE**, Goran Panic, Evangelos B. Mazomenos,.JosyAchner, Jasmin Klemke, Michael ,"**Low-complexity Framework for Movement. Classification** using Body-worn Sensors."IEEE transaction on VLSI 2017...

[2]. D. Biswas and K. Maharatna, "A CORDIC-based low-power statistical feature computation engine for WSN applications," Circuits, Syst.,Signal Process., vol. 34, no. 12, pp. 4011–4028, 2015.

[3]. Attal, Ferhat; Mohammed, Samer; Dedabrishvili, Mariam; Chamroukhi, Faicel; Oukhellou, Latifa; Amirat, Yacine. 2015. "**Physical Human Activity Recognition Using Wearable Sensors.**" **Sensors** 15, no.

[4]. L. Vachhani, K. Sridharan, and P. K. Meher, "Efficient CORDIC algorithms and architectures for low area and high throughput imple-mentation," IEEE Trans. Circuits Syst. II, Express Briefs, vol. 56, no. 1,61–65, Jan. 2009.

[5]. P. K. Meher, J. Valls, T.-B. Juang, K. Sridharan, and K. Maharatna, "50 years of CORDIC: Algorithms, architectures, and applications," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 56, no. 9, pp. 1893–1907,Sep. 2009.

[6]. J. Parkka, M. Ermes, P. Korpipaa, J. Mantyjarvi,J. Peltola, and I. Korhonen, "Activity classification using realistic data from wearable sensors," IEEE Trans. Inf. Technol. Biomed., vol. 10, no. 1, pp. 119–128, Jan. 2006.
[7]. F. E. Martínez-Pérez, J. Á. González-Fraga, J. C. Cuevas-Tello, and M. D. Rodríguez, "Activity inference for ambient intelligence through handling artifacts in a healthcare environment," Sensors, vol. 12, no. 1, pp. 1072–1099, 2012.
[8]. L. Wang, T. Gu, X. Tao, and J. Lu, "A hierarchical approach to real-time activity recognition in body sensor networks," Pervasive MobileComput., vol. 8, no. 1, pp. 115–130, Feb. 2012.
[9]. T. Rault, A. Bouabdallah, Y. Challal, and F. Marin, "A survey of energy-efficient context recognition systems using wearable sensors for healthcare applications," Pervasive Mobile Comput., Aug. 2006. [Online]. Available:
[10]. N. Wang, G. V. Merrett, R. G. Maunder, and A. Rogers, "Energy and accuracy trade-offs in accelerometry-based activity recognition," in Proc.22nd Int. Conf. Comput. Commun. Netw., Nassau, Bahamas, Jul. 2013,1–6.

SupriyaKurlekar "High Speed Algorithm for Arithmetic Operations Performed For Feature Extraction in Activity Recognition "International Journal of Engineering Science Invention (IJESI), vol. 07, no. 05, 2018, pp 47-55