IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 11, November 2016

FPGA Implementation of Cordic Processor for Square Root Function

Ms. Preeya Ambulkar[1], Prof. Ashish B. Kharate[2]

M.E. Student, Electronic & Telecommunication, H.V.P.M's College of Engineering & Technology, Amravati, India [1]

Asst. Professor, Electronic & Telecommunication, H.V.P.M's College of Engineering & Technology, Amravati, India[2]

Abstract: Today most of the DSP applications are supported real time transmission method. Digital illustrations of transmission data are typically handled inside a similar methodology as text; however the method rate possesses to be overabundant faster. On account of this real time production constraint, normal processors are not acceptable for up to date day DSP systems. Some hardware economical algorithms are, therefore required for these high speed applications. These algorithms need to be enforced associate optimized in hardware therefore on modify them to handle real time data whereas maintaining associate optimum trade-off between fully totally different performance parameters (speed and power). CORDIC is one such algorithmic rule. CORDIC (Coordinate Rotation Digital Computer) could also be a hardware economical shift-and-add algorithmic rule which is able to be wont to calculate varied arithmetic functions. The algorithmic rule incorporates a really simple operation requiring exclusively shift and add operations. So, this project aims to implement a CORDIC processor with every rotation mode and vectoring mode on FPGA Virtex-5. This project focuses on reducing low power in bit-parallel unrolled CORDIC structures by modelling the shift activity and the charging/discharging capacitance among the essential path.

Keywords: FPGA, Virtex-5, sine cosine, square root function, LUT.

I. INTRODUCTION

For an extended time the sector of Digital Signal process has been dominated by Microprocessors. This will be primarily as a result of the availability designers with the advantages of single cycle multiply-accumulate instruction what is more as special addressing modes. Although these processors are low price and versatile they are relatively slow once it involves activity positive troublesome signal process tasks e.g. compression, data communication and Video method. Of late, quick advancements are created among the sector of VLSI and IC style. As a result special purpose processors with custom-architectures have return up. The next speed is achieved by these custom-built hardware solutions at competitive costs. To feature to this, various straightforward and hardware-efficient algorithms exist that map well into these chips and should be wont to enhance speed and suppleness whereas activity the required signals process tasks.

One such straightforward and hardware-efficient algorithmic rule is CORDIC, associate kind for Coordinate Rotation data processor, projected by Jack E Volder in1959. It completely was developed to exchange the analog resolver among the B-58 bomber's navigation computer. CORDIC uses exclusively Shift-and-Add arithmetic with table Look-Up to implement whole completely different functions. By making slight changes to the initial conditions and thus the LUT values, it'll be used to efficiently implement pure mathematics, Hyperbolic, Exponential functions, Coordinate Transformations etc. victimization constant hardware. Since it uses exclusively shift-add arithmetic, VLSI implementation of such associate algorithmic rule is unquestionably realizable. DCT algorithmic rule has various applications and is wide used for compression. Implementing DCT victimization CORDIC algorithmic rule reduces the quantity of computations throughout method, can increase the accuracy of reconstruction of the image, and reduces the chip area of implementation of a processor designed for this purpose. This reduces the general power consumption. FPGA provides the hardware surroundings during which dedicated processors will be tested for their practicality. They perform varied high-speed operations that cannot be accomplished by a straightforward microchip. The first advantage that FPGA offers is on-the-spot two programmability. Thus, it forms the best platform to implement and take a look at the practicality of a frenzied processor designed CORDIC algorithmic rule.

II. RELATED WORK

The want of the CORDIC processors is already patterned out in the introduction section. It is additionally cleared that it is extremely about to be higher and better within the future physical science. At present, it's finding its nice use in embedded processors and certain about to capture the overall purpose processors market terribly presently. Several execs and cons of the CORDIC processors are mentioned by the system designers. The subsequent section can means the evolution of the projected project's work a look until date: during this paper [6] by Jack E. Volder, the CORDIC computing technique is very

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 11, November 2016

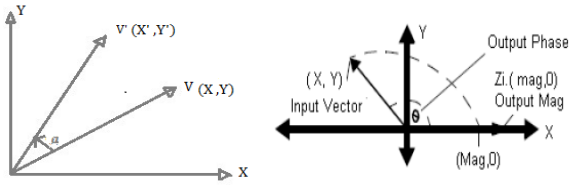
appropriate to be used during a special-purpose computer wherever the bulk of the computations involve pure mathematics relationships. During this paper [5] makes an attempt to survey exiting CORDIC and CORDIC like rule with an eye fixed towards implementation in FPGA. And shows that, is accessible to be used in FPGA based mostly in data processor, that area unit the doubtless basis for following generation DSP system. During this article, the completion of fifty years of the invention of CORDIC (COordinate Rotation DIgital Computer) by Jack E. Volder [6] [4], we tend to gift a short summary of the key developments within the CORDIC algorithms and architectures beside their potential and future applications. This paper [3] makes an attempt to explore the various implementations of CORDIC architectures, specific to FPGA devices. The algorithmic rule is enforced in 2 completely different styles: pleated and unrolled. Unrolled style is improved architecturally by pipelining it. Comparisons are then created between these architectures supported space, speed, turnout and power parameters and logical conclusions are drawn. All 3 styles are coded in VHDL and enforced victimization Xilinx FPGA synthesis tool. To examine the practicality of the algorithmic rule every of the styles has been simulated for perform / circular function} and circular function evaluations. During this paper [1][2] is that the inspiration for this project. It proposes promising resolution for top performance high speed and dynamic power dissipation. The focuses on reducing the power dissipation in bit-parallel unrolled CORDIC structures by modelling the switch activity and the charging/discharging capacitances among the important path. CORDIC is enforced to rotate the given coordinates (X, Y) [2] with the given angle θ and conjointly trigonometric function and circular function of given angle is found and waveforms are generated victimization Xilinx ISE tool. Examples are separate circular function Transformation (DCT) is employed for compression and Video Compression. DCT conjointly improves speed, as compared to alternative normal compression algorithms like JPEG. CORDIC are often used in communication for efficient generation of AM, modulation, PM, ASK, FSK, PSK, orthogonal frequency division multiplexing. As a result, combining the parallel and pipelined strategies of CORDIC processors can result in low power consumption with low latency of angle recording.

III. CORDIC ALGORITHM

The CORDIC is very straightforward and unvarying convergence algorithmic rule that reduces advanced multiplication, greatly simplifying overall hardware quality. this can be a reasonably option to system designers as they still face the challenges of feat aggressive price and power targets with the enlarged performance required in next generation signal process solutions. The basic principle underlying the CORDIC-based totally computation, and gifts its unvarying formula for numerous operational modes and two-dimensional organization. CORDIC algorithmic rule has two varieties of computing modes rotation and vectoring. The CORDIC methodology is often utilized in two different modes, namely, the rotation mode and also the vectoring mode. Within the rotation mode, the coordinate elements of a vector associated an angle of rotation are given, and the coordinate elements of the initial vector, after rotation through a given angle, are computed. Within the vectoring mode, the coordinate elements of a vector are given, and therefore the magnitude and angular argument of the initial vector are computed.

The CORDIC algorithmic rule performs the rotation of a vector in each mode as a sequence of micro-rotations by elementary angles [2] recalled from ROM. The number of micro-rotations for a given exactitude is decided by base getting used for the implementation of CORDIC algorithmic rule. The CORDIC is very straightforward and repetitive convergence formula that reduces difficult multiplication, greatly simplifying overall hardware quality. This is often a beautiful option to system designers as they still face the challenges of reconciliation aggressive price and power targets with the accumulated performance required in next generation signal method solutions. The essential principle underlying the CORDIC-based computation, and gift its repetitive formula for numerous operational modes and two-dimensional organisation.

CORDIC formula has 2 varieties of computing modes Vector rotation and vector translation. The CORDIC formula was initially designed to perform a vector rotation, where the vector V with components (X,Y) is turned through the angle θ yielding an action vector V' with part (X',Y') shown in Fig.1.



(a) (b)

Fig.-1. Vector rotation and translation

$$V' = [R]\,[V] \qquad (1)$$

Where **R** is the rotation matrix

$$R = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \qquad (2)$$

$$R = \begin{bmatrix} \frac{1}{\sqrt{1+\tan\alpha^2}} & -\frac{\tan\alpha}{\sqrt{1+\tan\alpha^2}} \\ -\frac{\tan\alpha}{\sqrt{1+\tan\alpha^2}} & \frac{1}{\sqrt{1+\tan\alpha^2}} \end{bmatrix} \qquad (3)$$

By factoring out the cosine term in (3), the rotation matrix **R** can be rewritten as

$$R = \left[1 + \tan^2\alpha^{-\frac{1}{2}}\right] \begin{bmatrix} 1 & -\tan\alpha \\ \tan\alpha & 1 \end{bmatrix} \qquad (4)$$

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 11, November 2016

And can be interpreted as a product of a scale-factor K= $[1+\tan^2\alpha^{-\frac{1}{2}}]$ with a pseudo rotation matrix, given by Rc

$$Rc = \begin{bmatrix} 1 & -\tan\alpha \\ \tan\alpha & 1 \end{bmatrix}$$

$$Rc = \begin{bmatrix} 1 & -\tan\alpha \\ \tan\alpha & 1 \end{bmatrix}$$

In vector translation, rotates the vector V with component (X, Y) around the circle until the Y component equals zero as illustrated in Fig. 2. The outputs from vector translation are the magnitude X' and phase θ', of the input vector V.

After vector translation, output equations are:

$$X' = Ki=\sqrt{x^2+y^2}$$
$$Y' = 0$$
$$\theta' = atan(\tfrac{y}{x})$$

To achieve simplicity of hardware realization of the rotation, the key ideas used in CORDIC arithmetic are to decompose the rotations into a sequence of elementary rotations through predefined angles that will be enforced with minimum hardware price and to avoid scaling, which can involve computing, like square-root and division. The second set up is based on the particular reality the scale-factor contains only the magnitude information but no information relating to the angle of rotation.
After few years, Walther found but CORDIC iterations can be modified to calculate hyperbolic functions and reformulated the CORDIC rule to a generalized and unified kind that is applicable to perform rotations in circular, hyperbolic and linear coordinate systems. The unified formulation includes a replacement variable m that is assigned fully totally different values for varied coordinate systems. The generalized CORDIC is developed as follows:

$$x_{i+1} = \{x_i - ny_i\sigma_i 2^{-1}\}$$
$$y_{i+1} = \{y_i + nx_i\sigma_i 2^{-1}\}$$
$$z_{i+1} = \{z_i - \tan^{-1} 2^{-i}\}$$

Here σi= $\{sign(zi)$ for rotation mode , $-sign(zi)$ for vectoring mode.}

3.1 Architecture

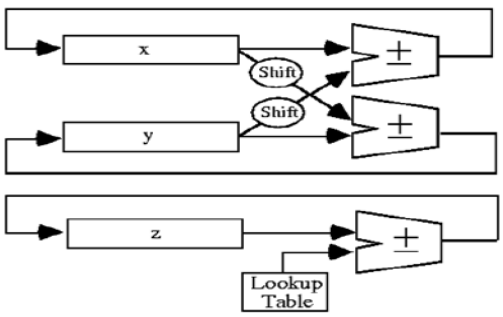


Fig.-2. Basic CORDIC processor

The above diagram explains the essential hardware design of a CORDIC processor. It shows the adders/ subtractor and also the shift registers. The adders/ subtractor perform the addition/subtraction of binary numbers. The register performs the bit-shift operation in accordance with the algorithmic rule. The constants equivalent to mounted angle values is obtained from the Look-up table enforced as a read-only memory. the present analysis within the style of high speed VLSI architectures for period of time digital signal process (DSP) algorithms has been directed by the advances within the VLSI technology, that have provided the designers with vital impetus for porting algorithmic rule into design. several of the algorithms utilized in DSP and matrix arithmetic need elementary functions like trigonometric, inverse trigonometric, logarithm, exponential, multiplication, and division functions. So, there are 2 varieties of architecture as given below.

a) Parallel CORDIC architecture

In this kind of design, all the iterations occur in a very single clock cycle. CORDIC algorithmic rule are enforced in a very kind of ways that within which. An instantaneous mapping of equations pattern in hardware leads to associate unvaried style. The unvaried style might even be either word-serial or bit-serial, relying on whether or not the practical unit implements the logic for one bit or for one word. The unvaried style has to perform iterations at n times the data rate. The unvaried structure are unrolled therefore that each of the n method elements frequently perform identical iteration. Unrolled architectures have two benefits; initial the shifters are designed for mounted shifts, which imply that they will be enforced among the wiring. Second, the store that holds the constant values for the z-branch need not to be updated once every iteration. These constants are hardwired rather than requiring hold. The entire CORDIC processor is so reduced to associate array of interconnected adder-subtraction units . The unrolled style are merely pipelined by inserting pipeline registers between the adder-subtraction units The design is as shown below in figure3

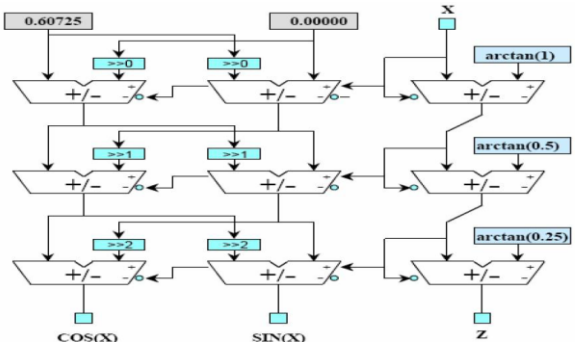


Fig.-3. Parallel CORDIC processor

b) Pipelined CORDIC architecture

As CORDIC iterations are identical, it is considerably convenient to map them into pipelined architectures. The most stress in economical pipelined implementation lies

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 11, November 2016

with the decrease of the essential path. Pipelined CORDIC circuits are used thenceforth for high-throughput implementation of curving wave generation, mounted and adaptative filters, distinct orthogonal transforms and alternative signal process applications. A generic design of pipelined CORDIC circuit is shown in Figure four. It consists of stages of CORDIC units wherever every of the pipelined stages consist of a basic CORDIC engine of the sort shown in Figure two. Since the quantity of shifts to be performed by the shifters at totally different stages is mounted (shift-operation through -bit positions is performed at the ordinal stage) just in case of pipelined CORDIC the shift operations can be hardwired with adders; and thus shifters are eliminated within the pipelined implementation. The critical-path of pipelined CORDIC so amounts to the time needed by the add/subtract operations in every of the stages. Pipelined design uses a structure kind of like that of a Parallel CORDIC. It uses pipeline registers in between every iteration section. Pipelined CORDIC proves to be advantageous with continuous input values.

For associate N bit information CORDIC core, N stage pipeline will offer most result. The primary output of associate N-stage pipelined CORDIC core is obtained when N clock cycles. Thereafter, outputs are generated throughout each clock cycle. The advantage of pipelined CORDIC core over parallel and unvaried CORDIC cores is its frequency of operation that is far higher when put next to the latter structures. Pipeline realizes same output as that of parallel core with improved frequency of operation. Downside of pipelined structure is that the increase in space introduced by the registers. Pipelined CORDIC implementation is neat in [1] and [4]. Hence, there is a trade-off between parallel and pipelined cores supported frequency and space. It is relatively the foremost economical CORDIC design. During this methodology multiple iterations occur in multiple clock cycles. It's enforced by inserting registers inside the various adder stages. The design is given as in Fig 4.

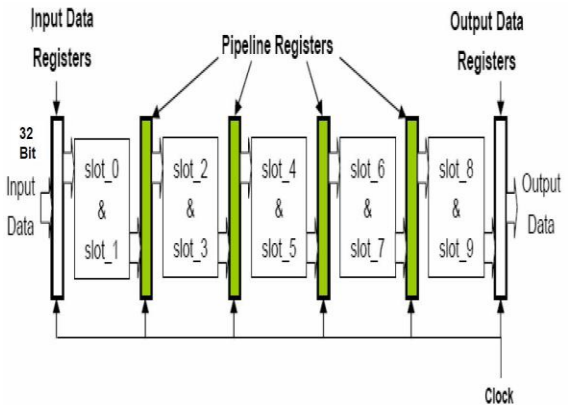


Fig.-4. Pipelined CORDIC processor

It has combinable circuit .It has extensive delay; however time interval is reduced as compared to the unvaried method. Shifters area unit of mounted size so may be enforced within the wiring. Constants may be hardwired rather than requiring space for storing. Parallel CORDIC may be pipelined by inserting registers between the adders stages. In most FPGA architectures there are a unit already registers gift in every logic cell, therefore pipeline registers has no hardware price. Range of stages when that pipeline register is inserted may be sculptured, considering clock frequency of system. Once in operation at larger clock amount power consumption in later stages reduces because of lesser change activity in every clock amount. All the higher than modules area unit to be synthesized exploitation Verilog HDL and enforced on FPGA.

c) Square root operation

The conversion of a floating-point range into base four victimization binary range representation is that the key part of the algorithmic rule approximation for computing the root of variety x. As explained in (2.1), by creating use of the base four and binary illustration, is simpler to form the number a part of the mantissa to be expressed in two bits mm.

$$v = MM.mmmm\ldots mm\ \sqrt{4^{exp}} \qquad (2.1)$$

As it are going to be elaborated additional on the subsequent chapters, the quantity of bits from the fractional half (fractional bits), mm....mm is decided according with the accuracy required on the input. For hard the root of the floating purpose range described in (2.1), the root is dead on the fixed-point part and also the exponent severally as it are going to be pictured within the equation (2.2), wherever the final output is called z

$$z = \sqrt{MM.mmmm\ldots mm}\sqrt{4^{exp}} = \sqrt{MM.mmmm\ldots mm}\ 2^{exp} \qquad (2.2)$$

The same approach as represented higher than, is used for computing the ordinal root, wherever n represent a number, as an example the cubical root of a floating purpose range. For the approximation of the rule for computing the cubical root of variety, x is adjusted into a floating purpose range with the base eight rather than four and binary illustration range as shown in2.3

$$v = MMM.mmmm\ldots..mm.\sqrt{8^{exp}} \qquad (2.3)$$

By exploitation the bottom eight and binary illustration of numbers, makes the number part of the fixed-point part to incorporates 3 bits MMM. As within the case of square root of variety, the quantity of fractional bits of the cubical root depends on the accuracy of the input also. within the same manner because the root, for performing the cubical root of the floating purpose numbers shown in (2.3), the fixed point and also the exponent are going to be thought of one by one as shown in 2.4

$$z = \sqrt[3]{MMM}.mmmm..mm.\sqrt[3]{8^{exp}} = \sqrt[3]{MMM}.mmmm..mm.2^{exp} \qquad (2.4)$$

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 11, November 2016

The allotment of bits in fixed-point is that an integer part, remaining fractional part and number of bits shift in computation.

i. Data Signals

The Data Signals are: X_IN, Y_IN, X_OUT and Y_OUT. For useful Configurations Rotate, Translate, Sin, Cos associated Atan the information Signals square measure represented victimization fixed-point 2's complement numbers with an number dimension of two bits. The number dimension is mounted despite the word width; the remainders of the bits square measure used for the fractional portion of the quantity input file signals, X_IN and Y_IN, should be within the range: -1 <= input file <= 1. Input file outside this vary produces undefined results.
Using a 10-bit word dimension, +1 and -1 square measure represented as:
"0100000000" => 01.00000000 => +1.0
"1100000000" => 11.00000000 => - 1.0

For the square root useful Configuration, the information Signals, X_IN and X_OUT, square measure each represented in either unsigned fractional or unsigned number data formatting. The input file signal, X_IN, should be within the vary:0 <= X_IN < +2 once data formatting is ready to unsigned Fraction or within the range 0 <= X_IN < 2**Input dimension once data formatting is ready to unsigned number. Once unsigned fractional data formatting has been designated the information Signals square measure represented employing an unsigned number with associate number with of one bit. The number dimension is mounted and also the remainder of the word is employed to represent the fractional portion of the quantity. victimization the System Generator Fix format this illustration is represented as UFix (N+1)_N, wherever is that the range of fractional bits getting used and is outlined as N = word dimension -1.The Q range format is employed to represent signed 2's complement numbers and is thus not appropriate to explain the illustration format employed by the root perform.

ii. Phase Signals

The phase Signals are: introduce and terminate. The phase signals square measure forever diagrammatic employing a fixed-point 2's complement range with associate number dimension of three bits. Like the information signals the number dimension is mounted and any remaining bits square measure used for the fractional portion of the quantity. The phase Signals need associate increased number dimension to accommodate the increased vary of values they have to represent once the phase Format is ready to Radians.
The output scaling is set as follows.
The CORDIC core calculates the root of input values within the vary0 <= X_IN < 2.
The CORDIC core calculates the root of input values within the vary0 <= X_IN < 2.
Y=√X
The alternative data formatting represents values within the vary0<= X_IN < 2N+1 and that we would like to calculate: Yalt=√Xalt
Interpreting Xalt victimization the quality CORDIC data formatting scales the input by 2-N,
Directly re-interpreting the CORDIC output within the various information formats leads to associate incorrect decimal price. This is often due the dimensions issue introduced by the remapping of the input and therefore the roots perform. This scaling issue introduced is shown on top of, 2-N/2. The corrected results are shown below:
UFix8_1 weighting: 16/2(6/2) = 2
UFix8_0 weighting: 32/2(7/2) = 2.8284
When N is even the scaling issue is associate number power of 2. This will be applied by merely right shifting the CORDIC output, X_OUT, by N/2. The instance victimization the UFix8_1 format demonstrates this with a scaling issue of 2-3 = 1/8.When N is odd the scaling issue is not associate number power of 2. This introduces a further output scaling issue of $\sqrt{2}$. The example victimization UFix8_0 demonstrates this with a scaling issue of 2-7/2 = 2-3.5.This could be enforced by 1st scaling the output by a right shift of four so multiplying by $\sqrt{2}$. An additional efficient method would be to translate the scaling to the input of the root perform. This is often demonstrated below where 2-N/2=2-M-(1/2).
The scaling becomes a straightforward divide by two, or right shift, of the input, X_IN, before applying it to the root perform. Followed by scaling the output, X_OUT, by 2-M.An input price of eight is employed for the UFix8_0 data format example. Divided by two this provides four. Four maps to 1/32 within the CORDIC input vary.
Sqrt (1/32) = 0.17678 = 0.0010110
It shows that the CORDIC output price, 0.0010110, maps to a decimal price of twenty-two in UFix8_0 data format. Applying the output scaling of 2-3, or 1/8, gives 2.75. The loss in accuracy is as a result of representing sqrt (1/32) victimization solely eight bits. If the complete accuracy result's used so re-interpreted to the alternative info (Fix8_0) so scaled, the proper result is obtained; for example:

Sqrt (1/32) * 27 * 2-3 = 2.8284

IV. RESULT

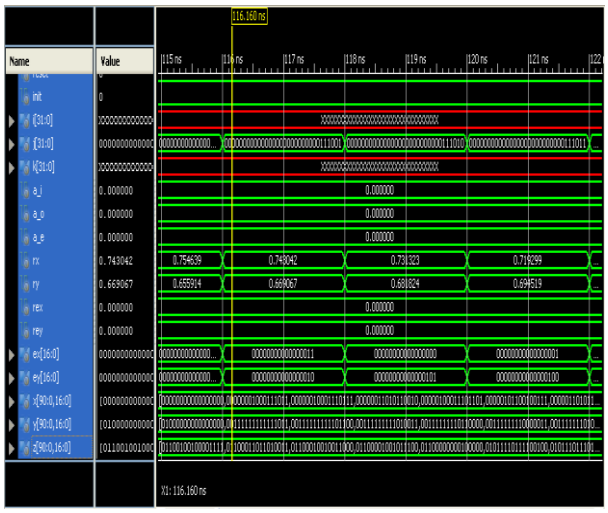
The implementation in this work is targeted FPGA families viz. Spartan-6 .The implementation is carried out for associate input quantity length varying from 16 bits. The style synthesis, mapping, translation and simulation are applied in Xilinx ISE 14.5. The trigonometric function uses simple pipelined architecture using CORDIC processor. The CORDIC is operated in rotating mode, hence only angle is given as input and x, y values are given in the program .As this architecture inputs can be given at every clock pulse and the value for inputs will output after eight clock cycle as it.
As it is observed, that CORDIC processors are going to expand their existence in the future high performance. This leads to lower scalability. Since the algorithm

IJARCCE

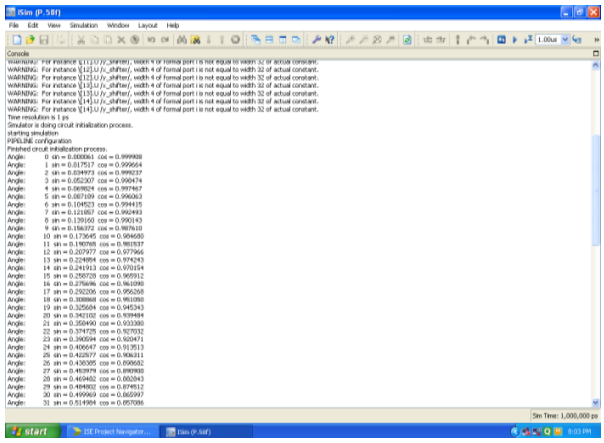
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 11, November 2016

involves only add and shift operations, it has very good hardware efficiency and a very minimal control overhead. The realization of this paper will solve most of the difficulties discussed above and in the problem definition section. This paper will have following results:



(a)



(b)

Fig. -5 (a) RTL Schematic and (b) console of sin – cos waveform CORDIC Processor

Fig – 6 RTL Schematic of square root waveform

CORDIC Processor

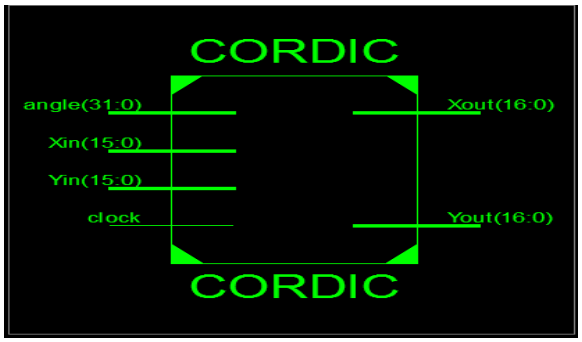
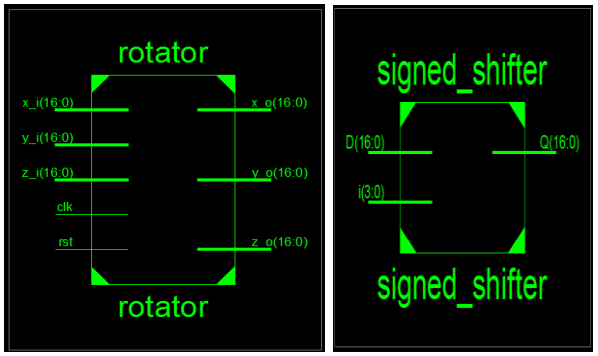
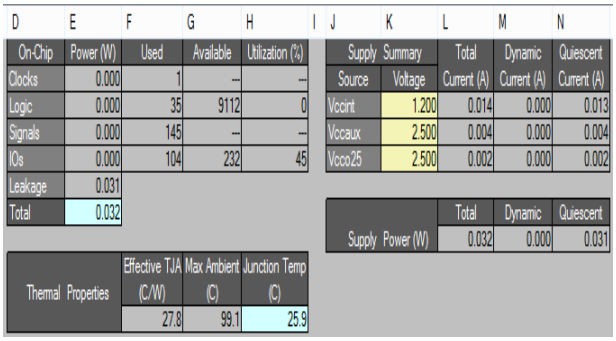


Fig. -7: RTL Schematic of CORDIC Processor



(a) (b)

Fig. -8 RTL Schematic of (a) rotator and (b) shifter



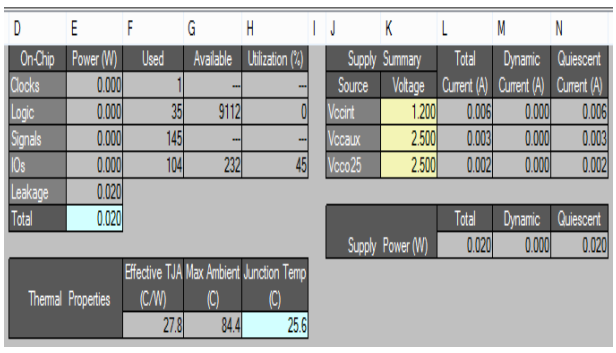
On-Chip	Power (W)	Used	Available	Utilization (%)
Clocks	0.000	1	--	--
Logic	0.000	35	9112	0
Signals	0.000	145	--	--
IOs	0.000	104	232	45
Leakage	0.031			
Total	0.032			

Supply Summary		Total	Dynamic	Quiescent
Source	Voltage	Current (A)	Current (A)	Current (A)
Vccint	1.200	0.014	0.000	0.013
Vccaux	2.500	0.004	0.000	0.004
Vcco25	2.500	0.002	0.000	0.002

	Total	Dynamic	Quiescent
Supply Power (W)	0.032	0.000	0.031

Thermal Properties	Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)
	27.8	99.1	25.9

Fig. -9 Power before optimization



On-Chip	Power (W)	Used	Available	Utilization (%)
Clocks	0.000	1	--	--
Logic	0.000	35	9112	0
Signals	0.000	145	--	--
IOs	0.000	104	232	45
Leakage	0.020			
Total	0.020			

Supply Summary		Total	Dynamic	Quiescent
Source	Voltage	Current (A)	Current (A)	Current (A)
Vccint	1.200	0.006	0.000	0.006
Vccaux	2.500	0.003	0.000	0.003
Vcco25	2.500	0.002	0.000	0.002

	Total	Dynamic	Quiescent
Supply Power (W)	0.020	0.000	0.020

Thermal Properties	Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)
	27.8	84.4	25.6

Fig. -10 Power from analyser

The results of power before optimization which is high and the power results from analyser which is low

DOI 10.17148/IJARCCE.2016.51115

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 11, November 2016

V. CONCLUSION

CORDIC is an intense calculation, and a thought calculation of call with regards to completely different Digital Signal process applications. Usage of a CORDIC-construct processor in lightweight of FPGA provides us associate intense element of execution complicated calculations on a stage that offers a large amount of assets and adaptableness at a moderately lesser expense. Further, behind the calculation is simple and effective the define and FPGA execution of a CORDIC based mostly processor is effortlessly accomplishable. CORDIC rule may be a crucial device that is employed as a region of Digital Signal process. The principle use of the CORDIC calculation during this paper is to arrange and make circular function and circular function values. This paper demonstrates that this calculation device is accessible to be used in FPGA based mostly process machines.

REFERENCES

[1] Burhan Khurshid & Roohie Naaz Mir Department of CSE National Institute of Technology – Srinagar, J&K India VLSI System, Architectures, Technology and Application(VLSI-SATA) **"**Power Efficient Implementation of Bit- Parallel Unrolled CORDIC Structures for FPGA Platforms" January 2015.

[2] A Ramya Bharathi & Mr. Md Masood Ahmad GITAM University Hyderabad "Rotation of Coordinates with Given Angle and to Calculate Sine/Cosine Using Cordic Algorithm"IEEE MAGAZINE volume no.2, issue no. :3 march 2015.www.ijmetmr.com

[3] Burhan Khurshid, Ghulam Mohd Rather & Hakim Najeeb-ud-din Department of Computer Science and software Engineering National Institute of Technology – srinagar, J&K India, "Performance Analysis of CORDIC Architectures Targeted for FPGA Devices" Volume 2, issue 2, February 2012 www.ijarcsse com

[4] Pramod K. Meher, Senior Member, IEEE, Javier Valls, Member, IEEE, Tso-Bing Juang, Member, IEEE,K. Sridharan, Senior Member, IEEE and Koushik Maharatna, Member, IEEE "50 Years of CORDIC algorithms, Architectures and Applications", IEEE 2009.

[5] R.Andraka, "A survey of CORDIC algorithms for FPGA based computers," FPGA '98, in ACM/SIGDA International Symposium on Field Programmable Gate Arrays, pp 191-200, 1998.

[6] J. E. Volder, "The CORDIC trigonometric computing technique," IRE Trans. Electronic computing, volume EC-8, pp 330 – 334, 1959.