Implementation Of CORDIC Algorithms For Fundamental Calculation

PRASHANT PATEL[1], HARDIK MODI[2]

[1] M.E. Student, VLSI and Embedded System Design, GTU PG School, Ahmedabad, Gujarat, India.
[2] Assistant Professor, CHARUSAT UNIVERSITY

ABSTRACT

CORDIC or Coordinate Rotation Digital Computer is a faster and easy to implement and powerful algorithm used in real time computations, trigonometric calculation. There are two mode in CORDIC Algorithm one is rotating mode and other is vectoring mode. Here We use CORDIC algorithm for fundamental calculation like triangular calculation using Xlinx 12.1 and FPGA Spartan 3E.CORDIC has use simple shift and add operation there is no multiplication and division is there. Here we use pipelined CORDIC algorithm for better performance. At the last Final results has been compared with base paper. Simple Multiplier is use to calculate simple mathematical calculation for more than two dimension and spectrum calculation generally not possible that's why here we design CORDIC algorithm.

Keyword: -*CORDIC, Dual Port RAM,FFT, Carry save adder*

1. INTRODUCTION

IN 1959 Volder has Design CORDIC algorithm for Bomber Navigastion.Microprossor and Microcontroller are temperature sensitive So that's why they do not give the perfect navigation that's why volder can design CORDIC algorithm for bomber navigation.CORDIC algorithms work in two mode one is vectoring mode and other is rotating mode.In Vectoring mode CO-ordinate component are given and the magnitude and angular of the original vector are computed.In the rotating mode co-ordinate component of a vector and angle of rotation are given and co ordinate component of original vector are computed. *Simple Multiplier is use to calculate simple mathematical calculation for more then two dimension and spectrum calculation genrally not possible that's why here we design CORDIC algorithm.IN CORDIC algorithms only shift and add operation occur no mulatipication is there so that's why iit's faster then other algorithms.Here we use pipelined CORDIC algorithms this algorithms is better then other algorims like sequentialand parallel…CORDIC algorithms work in both the coordinate system linear, circular and hyperbolic.Now aday CORDIC are Also use in Biomedical signal processing.The Rest of Paper is arranged as follows.In section 2 Problem statement and defination. In section III the proposed system.In section IV Implementation details and results.*

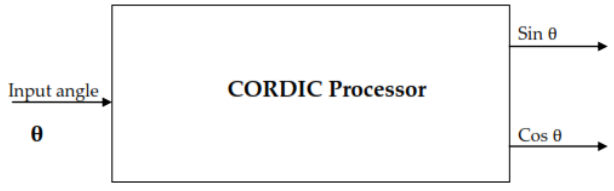

Fig. 1 Block Diagram of a CORDIC processor

II. PROBLEM STATEMENT AND DEFINITION

CORDIC is use in direct digital frequency synthesizer for sinne/cos generator.Here they use pipelined CORDIC architeture.Here CORDIC algorithms are use for sine/cose genretor.Here they use pipelined CORDIC algorithms that's why it's required large space compare to other algorithms.That's why this is not area sepecific application .[4]

IN This Paper They comper radix 2 and radix 4 CORDIC algorithms.After Impliment both the algorithms they conculed that radix 4 is better then radix 2 in latency.If Application is time requried then radix 4 CORDIC algorithms is better comper to radix 2[5].

In this paper there proposed system is area sepcific beacause here they use radix 2 algorithms and try to reduse memory area[8].

Now Here we work on both the parametter and make abetter system.

a. Time
b. Architecture
c. Complexity
d. Power Consumption

III. THE PROPOSED METHOD

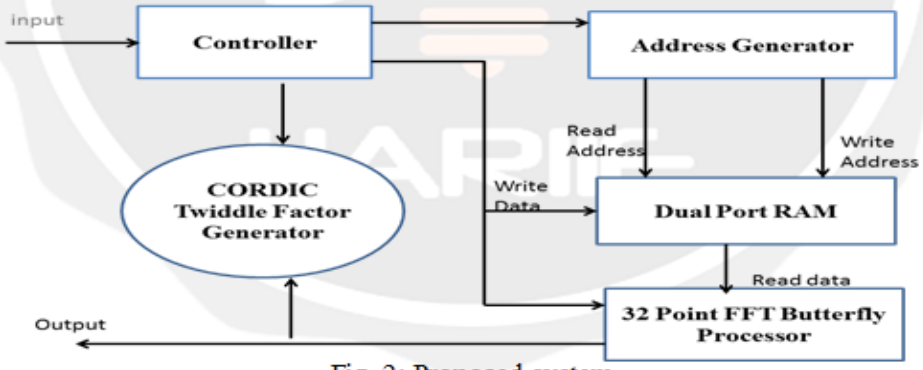

Fig. 2: Proposed system

First of all here we use pipelined CORDIC Algorithms, dual port ram, FFT twiddle factor, cpu control.The entire model is made of the address generation unit, the control unit, the dual port RAM unit, the 32-point butterfly unit and the CORDIC twiddle factor generation unit. In this proposed system I will use radix-4 pipelined CORDIC.Because In pipelined CORDIC architecture resister are connected between two adder and subtractor. After every process output and twiddle factor are store in that register. Controller can generate address of data and twiddle factor.

Pipelined CORDIC architecture:

You all aver with the word 'pipeline'.In pipelined architecture no of process can run at the same time.Here we work on FPGA, so In FPGA register are already present in every logic cell,there for no additionl hardwear is required and no additionl cost and give the better performance[14].

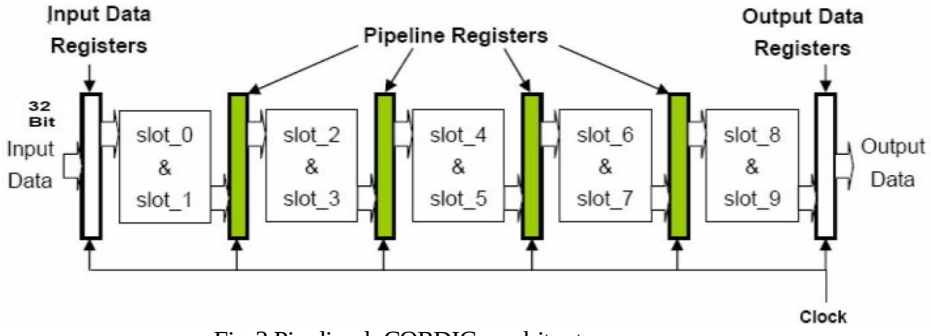

Fig 3:Pipelined CORDIC architecture

IV. IMPLEMENTATION DETAILS AND RESULTS

First of all I have check Adder report. which Adder is best for my algorithm

1. Ripple Carry Adder
2. Carry Save Adder
3. Carry Look Ahead Adder

Logic Utilization	**RCA**	**CLA**	**CSA**
No of Slices	58	44	50
Number of 4 Inputs LUTs	103	78	92
Number of Bonded IOBs	99	96	97
Delay	16.891ns	32.338ns	12.146ns

Table 1 Adder Reports

Here we implemented ripple carry adder, carry save adder, carry look ahead adder.And make a one table which compare both the adder in terms of delay and other perameter.After showing the results I have conclud that carry save adder has lowest delay compare to ripple carry adder and carry look ahead adder.Hear we use dual port RAM which is help to improved system performance.

CORDIC

The output of CORDIC module in this figure we can observe that CORDIC module takes X and Y as an input and gives Sine and Cos as output after computation. Amplitudes are defined for angle computation and every set of input is processed in tckhalf cycle and this result is forwarded to Dual port RAM for storage for later use. In each cycle one single input having real and imaginary term is multiplied with the twiddle factor. And final output will be derived.

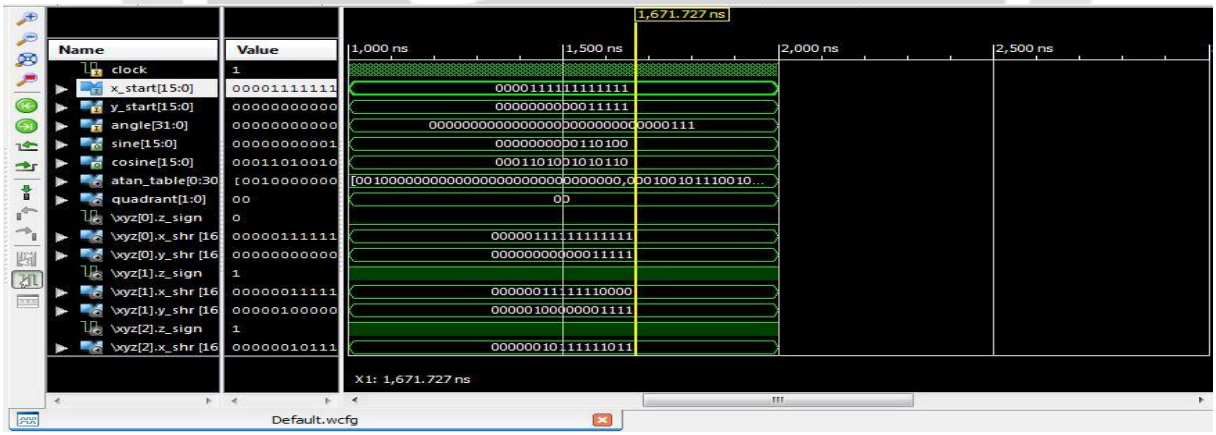

Fig 4: CORDIC Simulation Results

	BASE PAPER	OUTPUT
Number of slices	1274	981
Number of 4 input LUTs	2329	1841
Number of Bonded IOBs	189	97
Number of GCLKs	1	1

Table 2: Comparison With Base Paper

V. CONCLUSION

This paper design of CORDIC with 32 bit FFT Butterfly using HDL in Xilinx simulation. The output of the code shows and execute on different input angle. RTL and simulation results showed no error in code and also work correctly on different angle rotation. This CORDIC algorithm would be useful in many high-speed and real time applications.

V. REFERENCES

1. J. E. Volder, "The CORDIC trigonometric computing technique," IRE Transactions on Electronic Computers, vol. 8, no. 3, pp. 330–334, 1959.

2. Ron Warner, Cort Lansenderfer "Low cost, Low power FPGAs Provide Ideal Platform for CORDIC Applications" in lattice Semiconductor Corporation.

3. J. E. Volder, "The CORDIC trigonometric computing technique," IRE Trans. Electron. Computers, vol. EC-8, pp. 330–334, Sept. 1959.

4. P.Saravanan, S.Ramasamy, "Sine/Cos Generator For Direct Digital Frequency Synthesizer Using Pipelined Cordic Processor", 2013 IEEE.

5. Edwin Joseph, Rajagopal A. Karibasappa K, "FPGA Implementation of Radix-2 FFT Processor Based on Radix-4 CORDIC" 2014 IEEE.

6. M.D Puneet Bansal, B. S. Dhaliwal, S. S. Gill: Memory-efficient Radix-2 FFT Processor using CORDIC Algorithm. 2014 IEEE.

7. Sang Yoon Park, Member, IEEE, and Ya Jun Yu, Senior Member, IEEE," Fixed-Point Analysis and Parameter Selections of MSR-CORDIC With Applications to FFT Designs", Ieee Transactions On Signal Processing, Vol. 60, No. 12, December 2012.

8. Jian Wang, Chunlin Xiong, Kangli Zhang, and Jibo Wei, Member, IEEE, " Fixed-Point Analysis and Parameter Optimization of the Radix- 2^kPipelined FFT Processor", Ieee Transactions On Signal Processing, Vol. 63, No. 18, September 15, 2015

9. Vinay Kumar M, David Selvakumar A, Sobha P M, CDAC, Bangalore, " Area and Frequency optimized 1024 point Radix-2 FFT Processor on FPGA", 2015 International Conference on VLSI Systems, Architecture, Technology and Applications (VLSI-SATA)

.