High Speed Radix 8 CORDIC Processor

Smt. J.M.Rudagi[1], Dr. Smt. S.S ubbaraman[2]

[1]*Associate Professor, K.L.E CET, Chikodi, karnataka, India.*

[2]*Professor, W C E Sangli, Maharashtra.*

[1]js_itti@yahoo.co.in

***Abstract*: This paper presents high speed radix 8 CORDIC processor. Overall latency to estimate sine and cosine reduces 38% compared to that of radix 4 CORDIC processor. But hardware overhead increases due to additional adders and shifters required. The proposed design has been coded in Verilog and synthesized using Cadence RTL Encounter. PDP, EDP, and ADP parameters are estimated for radix 8 CORDIC processor and results are compared with respect to radix 4 CORDIC processor.**

Keywords: CORDIC, PDP, EDP, ADP.

I. INTRODUCTION

CORDIC the notion behind this computing machinery was motivated by the need to calculate the trigonometric and inverse trigonometric functions in real time navigation systems. The CORDIC requires only shift, add and table look up .with slight modification in initial condition, the core algorithm can multiply, divide and calculate square roots, hyperbolic, exponential and logarithmic functions.

J.E. Volder developed CO-ordinate Rotation Digital Computer (CORDIC) in 1959 to compute the rotation of two dimensional vectors [1]. Later Walther generalized this algorithm to compute logarithmic, exponential, division, hyperbolic and trigonometric functions [2]. CORDIC is an iterative algorithm for the calculation of the rotation of two dimensional vectors in linear, circular and hyperbolic coordinate systems. This rotation is carried out by a sequence of iterations. Each of this rotation over a prefixed elementary angle (micro rotation) is evaluated by means of addition and shift operations. The number of iteration of radix-2 CORDIC limits its architecture to use in high speed applications. Lot of work has been carried out to reduce the number of iteration to make it faster device. Recently there are various proposals for fast CORDIC algorithms. One of the most effective way to accelerate the CORDIC computation speed is to use redundant number of representations such as signed digits(SD) and carry save representation [9],[10]. Another method for rotation is prediction method. But these prediction methods make fast operation of basic CORDIC iteration step. However a number of prediction stages must be inserted among basic iteration stages, which sometimes causes irregular structure that are not suitable for VLSI implementation.

All the above approaches mentioned above does not mean reduction in the number of iterations itself. Hence it is likely that development of high Radix algorithm is essential for reducing the number of CORDIC iterations. It has been long been recognized that use of higher radices is effective for designing fast dividers. Thus radix 8 CORDIC processor has been presented in this paper.

The remainder of this paper proceeds as follows section 2 presents the Algorithm of radix 8 CORDIC processor. Section 3 covers the architecture of radix 8 CORDIC processor, simulation and synthesis results have been covered in section 4.

II. ALGORITHM

These are the basic equations of radix 8 CORDIC processor.

$$X(i+1) = X_i - \sigma_i 8^{-i} Y_i \qquad (1)$$

$$Y(i+1) = Y_i + \sigma_i 8^{-i} X_i \qquad (2)$$

$$Z(i+1) = Z_i - \sigma_i (\alpha_i) \qquad (3)$$

$$\alpha(i) = \tan^{-1} 8^{-i} \qquad (4)$$

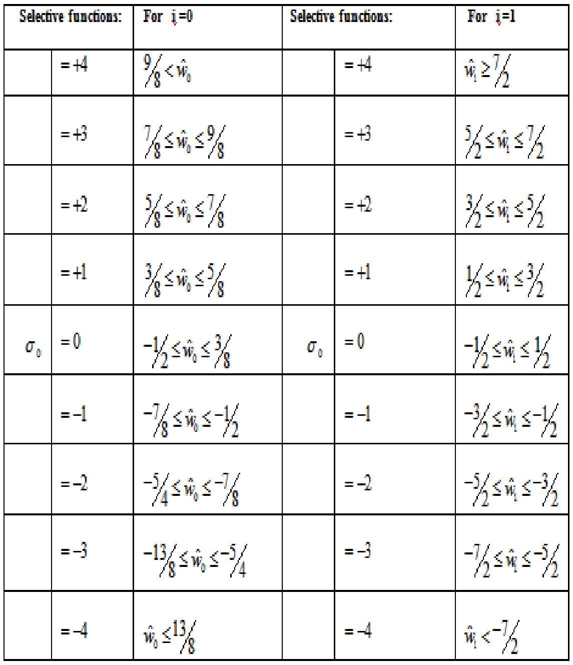
$$W_i = 8^i Z_i \qquad (5)$$

$$W(i+1) = 8^i \tan^{-1} (\sigma_i 8^{-i}) \qquad (6)$$

For higher speed and low power implementation redundant arithmetic is proper choice, which makes the addition carry free. These selection functions can be easily implemented in hardware.

The selection function obtained should be independent from the index of the iteration. So that the limits of intervals for selection i be same in every iteration. Then, the common overlapping area for all the iterations has to be identified, which determines the limits of selection intervals that are independent of i .This will be possible for micro rotation with i >0. Consequently, selection function for i > 0 and i = 0 is obtained separately. In order to make, implementation of the selection function in redundant arithmetic easy, value of $\boldsymbol{\sigma}_i$ must be determined by assimilating just a small number of most significant bits of w $_{i..}$The selection functions to calculate the value of σ is given below for iteration i =0, and i =1.is depicted in below table.

Table 1: Radix 8 selection function



Selective functions:		For i =0	Selective functions:		For i =1
	$=+4$	$9/8 < \hat{w}_0$		$=+4$	$\hat{w}_1 \geq 7/2$
	$=+3$	$7/8 \leq \hat{w}_0 \leq 9/8$		$=+3$	$5/2 \leq \hat{w}_1 \leq 7/2$
	$=+2$	$5/8 \leq \hat{w}_0 \leq 7/8$		$=+2$	$3/2 \leq \hat{w}_1 \leq 5/2$
	$=+1$	$3/8 \leq \hat{w}_0 \leq 5/8$		$=+1$	$1/2 \leq \hat{w}_1 \leq 3/2$
σ_0	$=0$	$-1/2 \leq \hat{w}_0 \leq 3/8$	σ_0	$=0$	$-1/2 \leq \hat{w}_1 \leq 1/2$
	$=-1$	$-7/8 \leq \hat{w}_0 \leq -1/2$		$=-1$	$-3/2 \leq \hat{w}_1 \leq -1/2$
	$=-2$	$-5/4 \leq \hat{w}_0 \leq -7/8$		$=-2$	$-5/2 \leq \hat{w}_1 \leq -3/2$
	$=-3$	$-13/8 \leq \hat{w}_0 \leq -5/4$		$=-3$	$-7/2 \leq \hat{w}_1 \leq -5/2$
	$=-4$	$\hat{w}_0 \leq 13/8$		$=-4$	$\hat{w}_1 < -7/2$

III. ARCHITECTURE:

The radix 8 CORDIC algorithm requires just 4 iterations to guarantee a 16 bit result. The radix 8 blocks is naturally more complicated than radix 4 blocks.

One radix block requires 4 adders, 4 shifters, 4 multiplexers. Since σ is no longer a function of two, so extra adders and shifters are needed to solve the case where $\sigma_i = \{3, -3\}$ .Multiplication by 3 is solved by left shift and an addition. Therefore radix 8 requires two more adders and shifters. The selection process is almost similar to same as that of radix 4.

The final architecture for radix 8 implementation is shown in figure below. The propagation delay decreases but cell area required increases .The adders are the Sign Digit Adders (SDA). The redundant adders are useful to decrease the overhead of propagating delay .It needs only more bits to represent the single binary digit. Therefore single digit has many representations. But when compared to non redundant it makes logical operations slower, but arithmetic operations are faster.

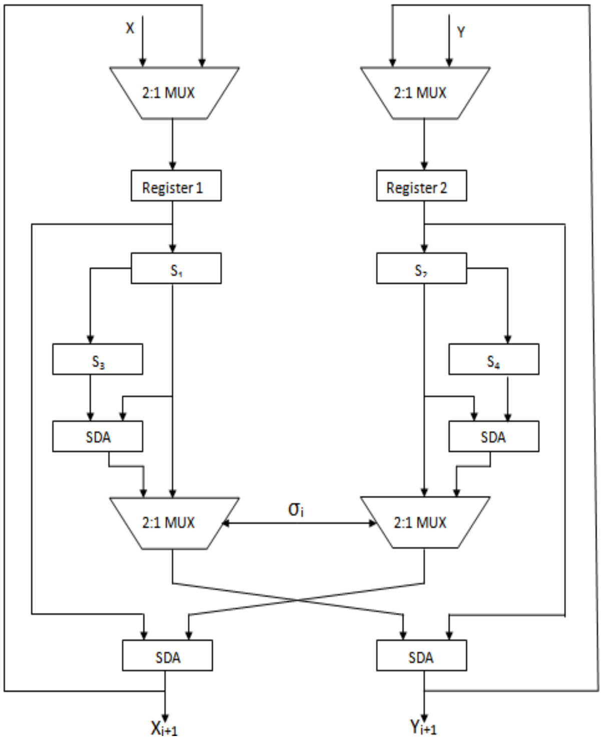


Fig.1: Radix 8 CORDIC architecture

IV. RESULTS AND DISCUSSION

The following table gives synthesis result in terms of call area, delay and power. Both the architectures have been modeled in VERILOG[15,16]. Then it has been synthesized using CADENCE RTL Encounter

Table 2 Synthesis Results for radix 8 CORDIC Processor

CORDIC Processor	Cell area	Delay(ns)	Leakage power(mW)	Dynamic power(mW)	Total power(mW)
Radix 4	3900	3.25	0.0288	0.1353	0.1641
Radix 8	5126	2.15	0.032	0.2001	0.2321

From the above table it can be concluded that there is a 6% decrease in PDP from Radix 4 to Radix 8. A lower PDP means that power consumption is better translated into speed of processor.

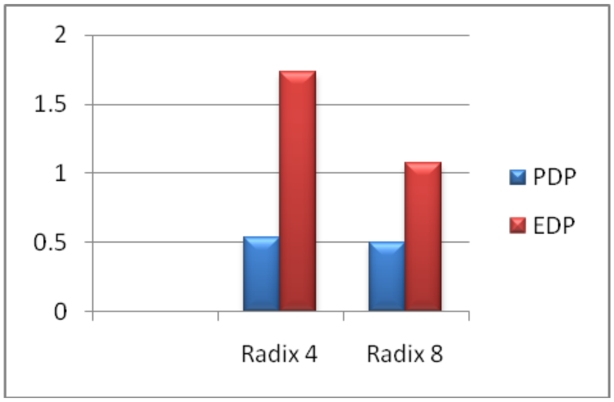


Fig.2: Performance analysis of Radix 4 and 8 CORDIC processor in terms PDP, EDP

EDP decreases 38% from Radix 4 to Radix 8.that shows energy saving of 38%. From Radix 4 to Radix 8, 10% decrease in ADP with 23% increase in area. When applications require high throughput and high speed, Radix 8 CORDIC is the best choice, as Radix 8 CORDIC throughput increases by around 60% compared to that of Radix 4.

V. CONCLUSION.

The efficient hardware supports is required for multiply and add operation for present day real time signal processing application. As CORDIC processors which makes use of only shift and add operations is best choice for VLSI implementation but latency is the main bottleneck in CORDIC processor, so latency can be reduced by using higher radix CORDIC processor even though hardware complexity increases.

ACKNOWLEDGMENT

The authors are immensely grateful to the valuable suggestions and help provided by the Department of Electronics and Communication Engineering, KLE College of Engineering and Technology, Chikodi, Belgaum, Karnataka.

. REFERENCES

1. J.Volder, " The CORDIC Trigonometric Computing Technique", IRE Trans On Electronic Computer, Vol, Ec8 No 3, pp330-334, Sept 1959.

2. J.S.Walther, ‟ A Unified Algorithm for Elementary functions", spring joint Computer Conference, Vol 38, pp37-385, 1971.

3. Y.H.Hu,‟ The Quantization Effects of the CORDIC algorithm" ,IEEE Transaction on Signal Processing, Vol 40,No 4,1992,pp 834-844.

4. N.Takagi,S.Yajima,‟ Redundant CORDIC method with constant scale factor for sine and cosine computations", IEEE Transaction Computers, Vol 40,No 9,pp 989-995,1991.

5. T.Aoki, H.Nagi, T.higuch, ‟ High Radix CORDIC algorithm for VLSI signal Processing", Proceeding 1997 IEEE workshop on Signal Processing Systems pp 183-192, Nov 1997.

6. Antelo, brugera, ‟Redundant CORDIC Rotator based on CORDIC algorithm", IEEE Transaction on computers, 1995.

7. J.D.Brugera, Antelo, E.L.Zapata, ‟ Design of pipelined radix 4 CORDIC processor", Parallel Computing, Vol 19, Nov 7, pp729-744, 1993.

8. J.Villaba,J.A.Hidalgo,Antelo,J.Dbrugera,‟ CORDIC Architectures With Parallel Compensation Of Scale Factor", In Proceedings Of The International Conference On Application Specific Array Processors,pp258-269,July 1995.

9. B. Lakshmi, A.S.Dhar ,‟Low Latency VLSI architectures for radix 4 CORDIC processor ", IEEE region 10 colloquium and Third International Conference on Industrial and Information Systems, Kharagpur , India, Dec 8-10,2008.

10. A. Gonzalez, P.Mazumdar, ‟Redundant arithmetic, algorithm and implementation".

11. Parhi, K. K., "VLSI Digital Signal Processing Systems: Design and Implementation", Wiley, 1999.

12. Hwang, K., "Computer Arithmetic: Principles, Architectures and Design", Wiley, 1979.

13. Guyot, A., Herreros, Y., Muller, J., "JANUS, an on-line multiplier/divider for manipulating large numbers", in Proc. of 9th Symposium on Computer Arithmetic, pp. 106 – 111, 1989.

14. "ISE Simulator", Xilinx incorporation San Jose U.S.A, 2011.

15. J.M.Rudagi, S .Subbaraman, "*Performance analysis and FPGA implementation of radix 2 and Radix 4 CORDIC"*, International Journal of Engineering, Science and Innovation Technology, ISSN No: 2319-5967(ISO: 9001:2008 Certified).

16. J.M.Rudagi, S.Subbaraman, "FPGA implementation of Redundant CORDIC processor", First international conference on Recent Innovation in Engineering and Technology, Bhalki, karnataka, India 11th -12th Nov 2016.

17.