Contemporary Engineering Sciences, Vol. 7, 2014, no. 9, 435 - 441
HIKARI Ltd, www.m-hikari.com
http://dx.doi.org/10.12988/ces.2014.4327

A Modified Scale Free CORDIC Architecture to Improve the Speed

M. Raguram and R. Parameshwaran

SASTRA University, India

Abstract

In spite of several modified versions proposed, there is a huge demand for an efficient implementation of CORDIC in signal processing applications. Various methods proposed for CORDIC have their own drawbacks either in range of functionality or complexity. This paper presents speed optimized scale-free CORDIC in order to be mainly used in signal processing applications such as FFT and DCT etc.The proposed CORDIC provides enough flexibility to be implemented in signal processing applications to carry out vector rotation operation with required accuracy. The system uses less number of slices with increase in speed when compared with conventional version upon implementing on Xilinx virtex 5. The sine and cosine waveforms of the pipelined design is also shown for verification.

Keywords: cosine/sine, scale-free CORDIC, pipelined architecture, vector rotation

1. Introduction

The coordinate rotation digital computer [6] has found applications [5] in many areas such as RADAR signal processing, bomber jets, math coprocessors, hand held calculators and software defined radios for generating sine and cosine functions, calculation of several transforms such as discrete sine/cosine transforms, Fast Fourier transforms etc. Several modified versions of CORDIC has been proposed in order to reduce the complexity of conventional algorithm to improve the speed and area. The virtually scaling [3], [4] free has been proposed with increased area considerably in order to acquire desired region of convergence. The enhanced scale free version [2] increases the region of convergence using hybrid method containing both conventional and scale free

version which in turn degrades the performance. The area-time efficient CORDIC [1] overcomes the scaling problem completely but with reduced speed causing delay of seven clock cycles for output to be pushed out for each input given. This paper presents a speed optimized version of CORDIC that is flexible enough with operation in entire coordinate space to be applied to any signal processing application that requires computation of rotation. This article is organised into different sections. Section 2 displays the review of existing CORDIC algorithms. Section 3 describes the proposed architecture and section 4 presents the field programmable gate array implementation. Section 5 deals with comparison of results with existing architecture. Section 6 concludes the paper.

2. Concise review of existing CORDIC algorithms

2.1 Conventional algorithm

Conventional CORDIC has been designed in an attempt to reduce the complexity of conventional rotation operation by eliminating the use of multipliers in hardware. In this the rotation operation is performed by iterative elementary rotations. The equation for finding intermediate elementary rotation of next iteration is given by

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \mathrm{k}_i \cdot \begin{bmatrix} 1 & -\tan a_i \\ \tan a_i & 1 \end{bmatrix} \cdot \begin{bmatrix} x_i \\ y_i \end{bmatrix} \tag{1}$$

Where $\mathrm{k}_i = \cos a_i$

$$a_i = \tan^{-1}(2^{-i}) \tag{2}$$

Since the total number of iterations is equal to the word-length (b) of the inputs, the rotation theta is given by $\theta = \sum_{i=0}^{b} a_i . \mu_i$ (3) where $\mu_i \in \{1,-1\}$ depending on the direction of rotation. In conventional CORDIC even though the multipliers are eliminated, it still has a complexity equal to a multiplier. This because of the k value that is needed to be multiplied at the end of the last iteration in order to get the exact result. The value of k depends on the number of iterations. It is given by

$$\mathrm{K} = \prod_{i=0}^{b} \mathrm{k_i} = \prod_{i=0}^{b} 1/\sqrt{1+2^{-2i}} \tag{4}$$

2.2 Existing scaling free versions

Initially attempts were made to eliminate the scaling operation at the end of iterations. So first order sine and cosine series were used

$$\sin a_i = 2^{-i} \quad \text{and} \quad \cos a_i = 1 - 2^{-(2i+1)} \tag{5}$$

However this version suffered with very low convergence range because of the restriction that the basic shift should be greater than a certain value.

To overcome this problem modified virtually scaling free algorithm has been proposed but still it needs adaptive scale factor to be multiplied in order to make the convergence to entire coordinate space. In enhanced scaling free CORDIC radix 4 booth recoding is used along with conventional CORDIC which is likely less flexible to implement in other applications because of its complexity.

Taylor's series is used for approximating sine and cosine functions where the sine and cosine functions are given by

$$\sin a = a - (3!)^{-1}.a^3 + (5!)^{-1}.a^5 \ldots \tag{6}$$

$$\cos a = 1 - (2!)^{-1}.a^2 + (4!)^{-1}.a^4 \ldots \tag{7}$$

In the above version the value of basic shift varies depending on the order of Taylor's series and input word-length. It is given by

$$\text{Basic shift} = \frac{b - log_2(n+1)!}{(n+1)} \tag{8}$$

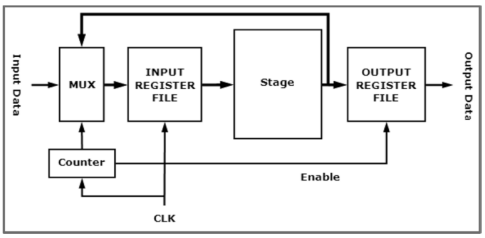

Figure 1. Block diagram of existing system

In this version using an algorithm the value for the angle of next iteration along with the shift value for next iteration has been calculated. The total number of iterations is equal to the sum of number of iterations done with basic shift and the number of iterations done with shifts for each iteration value. In this version speed has been traded-off in order to reduce the area. So recursive architecture has been proposed, the practical implementation of which is somewhat complex in terms of synchronisation.

3. Proposed architecture:

In the proposed architecture critical path has been reduced considerably by a pipelined version which is also feasible for implementation in signal processing applications. It reduces the complexity in terms of synchronisation and also overcomes the problem of considerable delaying of output to be produced after every input is given which exists in the existing version.

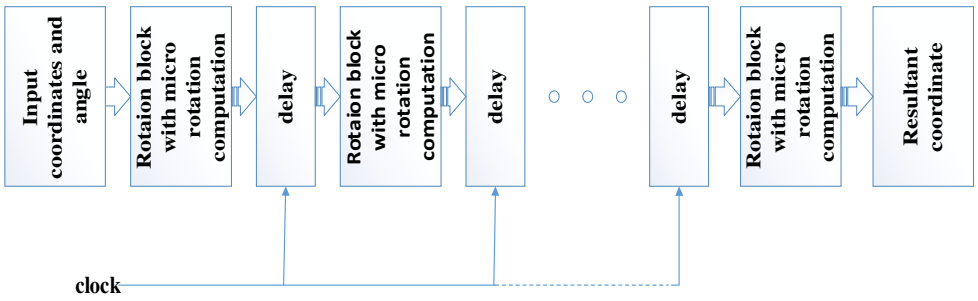

Figure 2. Seven stage pipelined architecture

The proposed architecture eliminates the use of counter, multiplexer, input and output registers needed for synchronisation as in existing architecture. In addition it is capable of delivering the output continuously after every clock cycle for series of inputs given in spite of initial latency. In order to pipeline, delays are added between multiple copies of rotation block for the purpose of reducing the critical path that exists in the previous version of scale free CORDIC architecture.

In the proposed architecture third order Taylor's series approximation of CORDIC with input bit width of 16 has been implemented, which requires only seven iterations. So seven stage pipelined version of scale free CORDIC has been implemented. The sine and cosine functions are approximated as follows

$$\sin a = a - \frac{a^3}{3!} \tag{9}$$

$$\cos a = 1 - \frac{a^2}{2!} \tag{10}$$

Implementing the above equations with the approximation $a = 2^{-si}$ in the rotation equation

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} 1 - \frac{2^{-2si}}{2} & -(2^{-si} - \frac{2^{-3si}}{2^2}) \\ 2^{-si} - \frac{2^{-3si}}{2^2} & 1 - \frac{2^{-2si}}{2} \end{bmatrix} \cdot \begin{bmatrix} x_i \\ y_i \end{bmatrix} \tag{11}$$

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} 1 - 2^{-(2si+1)} & -(2^{-si} - 2^{-(3si+3)}) \\ 2^{-si} - 2^{-(3si+3)} & 1 - 2^{-(2si+1)} \end{bmatrix} \cdot \begin{bmatrix} x_i \\ y_i \end{bmatrix} \tag{12}$$

Here si is shift value calculated at every iteration.

The algorithm for calculating micro rotation and shift value is shown in figure 3.1.

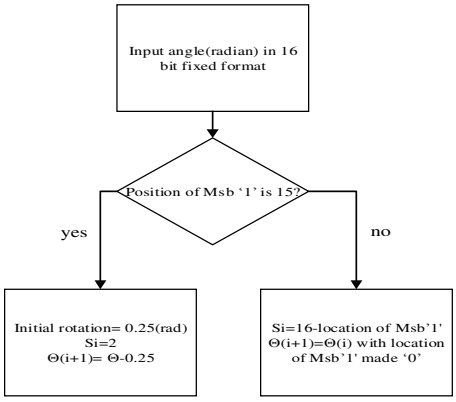

Figure 3.1 Flow chart for calculating micro rotation architecture

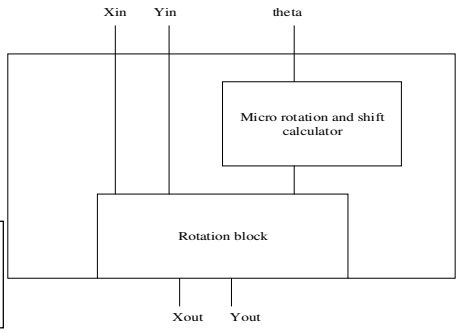

Figure 3.2. Single stage in a pipelined

Each stage in the pipelined architecture consists of the micro rotation calculation block along with the rotation block. Figure 3.2.

4. Field programmable gate array implementation

The proposed architecture has been designed using Verilog HDL and synthesized in Xilinx 14.7 for implementation in virtex 5.The RTL schematic of the implemented architecture is shown in the figure 4. For the comparison of the no. of slices used in the proposed architecture with pipelined version of conventional CORDIC, both the architectures are implemented in virtex 5 in order to obtain uniformity.

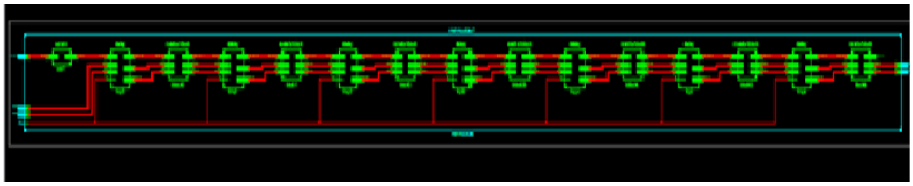

Figure 4. RTL schematic of the synthesized design.

5. Comparison of results

The pipelined version of conventional CORDIC uses 914 slices whereas the proposed pipelined scale free architecture uses only 325 slices.

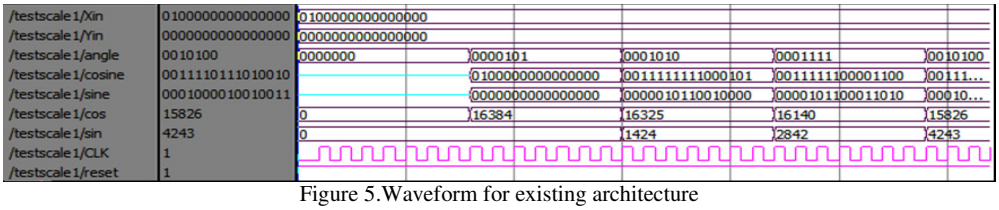

Figure 5.Waveform for existing architecture

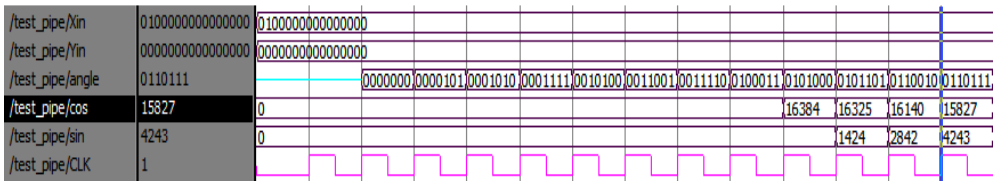

Figure 6. Waveform for proposed architecture

The waveform for sine and cosine function calculation of both the existing and proposed versions of scale free architecture in 16 bit fixed point format are shown in the figure 6 and 7 respectively.

6. Conclusion

The proposed architecture is capable of pushing outputs continuously at every next clock cycle for series of inputs given whereas the existing version causes the output to be delayed by seven clock cycles for every input given. The proposed design has only the initial latency of seven clock cycles for the first output to be pushed out. Therefore the proposed architecture is fast enough to be implemented in signal processing applications.

References

[1] S. Aggarwal, P. K. Meher, and K. Khare, "Area-Time Efficient Scaling-Free CORDIC Using Generalized Micro-Rotation Selection," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 8, pp. 1542–1546, Aug. 2012.

[2] F. J. Jaime, M. A. Sanchez, J. Hormigo, J. Villalba, and E. L. Zapata, "Enhanced Scaling-Free CORDIC," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 7, pp. 1654–1662, Jul. 2010.

[3] K. Maharatna, S. Banerjee, E. Grass, M. Krstic, and A. Troya, "Modified virtually scaling-free adaptive CORDIC rotator algorithm and architecture," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 11, pp. 1463–1474, Nov. 2005.

[4] Maharatna, K.; Troya, A.; Banerjee, S.; Grass, E., "Virtually scalingfree adaptive CORDIC rotator," *Computers and Digital Techniques, IEEE Proceedings,* vol.151, no.6, pp.448,456, 18 Nov. 2004.

[5] P. K. Meher, J.Walls, 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. E. Volder, "The CORDIC trigonometric computing technique," *IRE Trans. Electron. Comput.*, vol. EC-8, pp. 330–334, Sep. 1959.

[7] C.-S.Wu, A.-Y.Wu, and C.-H. Lin, "A high-performance/low-latency vector rotational CORDIC architecture based on extended elementary angle set and trellis-based searching schemes," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 50, no. 9, pp. 589–601, Sep. 2003.

Received: March 1, 2014