



Area Time Efficient scaling free Rotation modeCORDIC using Circular trajectory

DR.P HUNDEEP, Associate Professor, Department of CSE DHURVA INSTITUTE OF ENGINEERING & TECHNOLOGY, HYDERABAD, hundeepsurya11@gmail.com.

DR.G SRIVIDYA, Associate Professor, Department of CSE ELLENKI COLLEGE OF ENGINEERING & TECHNOLOGY, HYDERABAD, Veerabhadrasrividya@gmail.com.

DR.T SUNEEL, Professor, Department of CSE DHURVA INSTITUTE OF ENGINEERING & TECHNOLOGY, HYDERABAD, suneelkumarvarma2013@gmail.com.

ABSTRACT

This paper presents an area-time efficient Coordinate Rotation Digital Computer (CORDIC) algorithm that completely eliminates the scale-factor. Besides we have proposed an algorithm to reduce the number of CORDIC iterations by increasing the number of stages. The efficient scale factor compensation techniques are proposed which adversely effect the latency/throughput of computation. The proposed CORDIC algorithm provides the flexibility to manipulate the number of iterations depending on the accuracy, area and latency requirements. The CORDIC is an iterative arithmetic algorithm for computing generalized vector rotations without performing multiplications.

Index Terms—coordinate rotation digital computer (CORDIC), cosine/sine, field-programmable gate array (FPGA), most- significant-1, recursive architecture.

I.INTRODUCTION

Year 2009 marks the completion of 50 years of the invention of CORDIC (Coordinate Rotation Digital Computer) by Jack E. volder. The beauty of CORDIC lies in the fact that by simple shift-add operations, it can perform several computing tasks such as the calculation of trigonometric, hyperbolic and logarithmic functions, real and complex multiplications, division, square-root and many others. The CORDIC is an entire-transfer computer, it contains a special arithmetic unit consisting of three shift registers, three adder- subtractor, and special interconnections. The CORDIC is applied in diverse areas such as signal and image processing, communication systems, robotics and 3-D graphics etc[1]-[3]. For applications where the angle of rotation is known in advance, a method to speed up the execution of the CORDIC algorithm by reducing the total number of iterations is presented. This is accomplished by using a technique called angle recoding. The proposed MVR- CORDIC algorithm (modified vector rotational CORDIC) will saves the 50% execution time in the iterative CORDIC structure, or 50% hardware complexity in the parallel CORDIC structure compared with the conventional CORDIC scheme [4]-[6]. The corresponding architectures come for both rotation and vector modes and the other only for rotation mode to perform the scaling factor compensation in parallel with the classical CORDIC iterations. For fixed point arithmetic area and latency of the proposed implementation is

compared with standard CORDIC[7]-[8]. The two area - time efficient CORDIC architectures have been suggested in [9]. In

[11] the Coordinate Rotation Digital Computer (CORDIC) rotator is a well known and widely used algorithm within computers due to its way of carrying out some calculations such as trigonometric functions, many others. The new architecture which are able to reach a 35% lower latency and a 36% reduction in area and power consumption compared to the original scaling free architectures.

II. BRIEF OVERVIEW OF CORDIC ALGORITHM

To evaluate trigonometric functions we have many approaches such as 1) Polynomial Approximations
2) Table lookup
3) CORDIC

1) Taylor Series

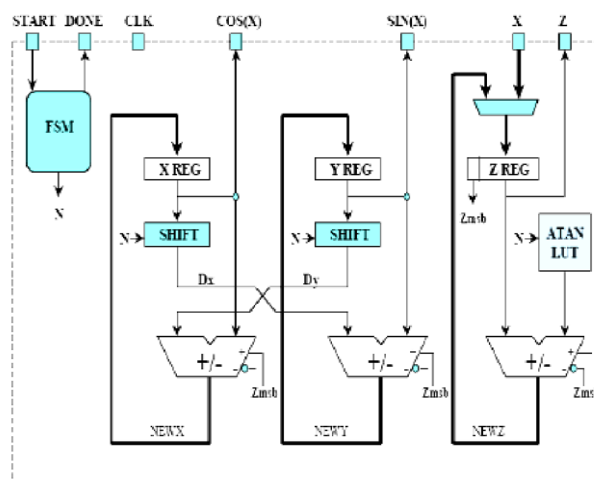
The Taylor series expansion for sine is:

$$x^3 + \frac{x^5}{5!} + \frac{x^7}{7!} \sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

This method is one of the oldest and most widely, but the problem associated with this method is, to get values of higher accuracies, higher order factorial and power has to be calculated. Moreover to implement this we would at least require a multiplier, divider, adder and a subtractor. For good accuracy it would be required to take each term in calculation till they become insignificant. Thus this approach has a lot of hardware requirements as well as it is slow.

2) Look up Table

The Look up table approach involves storing values of sine and cosine at different angles. Based on the number of values stored, the lookup table can be big or small, but clearly, the smaller the lookup table, more is the error involved. The problem with a bigger lookup table is that it requires more memory and memory is expensive. Moreover the size of the Lookup table increases exponentially with the increase in the precision of the angle. Though this approach provides fast results it is very expensive to implement.



3) Cordic Algorithm

CORDIC is an acronym for Coordinate Rotation Digital Computer introduced by Jack E. Volder. It is an iterative algorithm capable of calculating trigonometric and various other functions. In this algorithm with the help of an adder/subtractor, a small look up table and a shifter the trigonometric functions can be calculated very easily. The advantage that Cordic offers over other algorithms are that it does not require multiplication or division blocks, instead it works only with a shifter, adder/subtractor and a small lookup table. This reduces the hardware requirement drastically and provides reasonably good speed.

Many variations have been suggested for efficient implementation of CORDIC with less number of iterations over the conventional CORDIC algorithm [4]–[11]. The number of CORDIC iterations are optimized in [4]–[6] by greedy search at the cost of additional area and time for the implementation of variable scale-factor. In [7] and [8] efficient scale-factor compensation techniques are proposed, which adversely affect the latency/throughput of computation. Two area-time efficient CORDIC architectures have been suggested in [9], which involve constant scale-factor multiplication for adequate range of convergence (RoC). The virtually scale-free CORDIC in [10] also requires multiplication by constant scale-factor and relatively more area to achieve respectable RoC. The enhanced scale-free CORDIC in [11] combines few conventional CORDIC iterations with scaling-free CORDIC iterations for an efficient pipelined CORDIC implementation with improved RoC. However, if used for recursive CORDIC architecture, combining two different types of CORDIC iterations, degrades performance.

The low complexity technique for eliminating the scale factor is the use of Taylor series expansion. The Scaling-Free CORDIC and modified scale-free CORDIC are techniques based on Taylor series approach. The former suffers from low range of convergence (RoC) which renders it unsuitable for practical applications, while the latter extends the RoC but introduces predictable but constant scale-factor of $1/\sqrt{2}$. The other hardware efficient architectures require scale-factor compensations to extend the range of convergence to the entire coordinate space.

III. SEQUENTIAL/ITERATIVE CORDIC

It requires Maximum number of Clock Cycles to calculate output, Minimum Clock Period per iteration, Variable Shifters do not map well on certain FPGA's due to high Fan-in.

Fig.1. Variable Shifters

Parallel/Cascaded CORDIC:

It has Combinational circuit More Delay, but processing time is reduced as compared to iterative circuit. Shifters are of fixed shift, so they can be implemented in the wiring. Constants can be hardwired instead of requiring storage space.

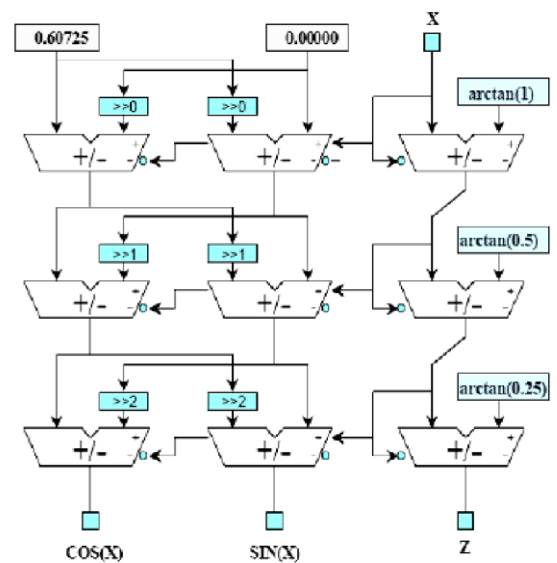


Fig.2. Parallel/Cascaded CORDIC

The key concept of CORDIC arithmetic is based on the simple and ancient principles of two-dimensional geometry. But the iterative formulation of a computational algorithm for its implementation was first described in 1959 by Jack E. Volder for the computation of trigonometric functions,

multiplication and division. This year therefore marks the completion of 50 years of the CORDIC algorithm. Not only a wide variety of applications of CORDIC have emerged in the last 50 years, but also a lot of progress has been made in the area of algorithm design and development of architectures for high performance and low-cost hardware solutions of those applications. CORDIC-based computing received increased attention in 1971, by varying a few simple parameters; it could be used as a single algorithm for unified implementation of a wide range of elementary transcendental functions involving logarithms, exponentials, and square roots along with those suggested by Volder. During the same time, Cochran benchmarked various algorithms, and showed that CORDIC technique is a better choice for scientific calculator applications. The popularity of CORDIC was very much enhanced thereafter primarily due to its potential for efficient and low-cost implementation of a large class of applications which include: the generation of trigonometric, logarithmic and transcendental elementary functions; complex number multiplication, eigen value computation, matrix inversion, solution of linear systems and singular value decomposition (SVD) for signal processing, image processing, and general scientific computation.

The name CORDIC stands for Coordinate Rotation Digital Computer. Volder [Vold59] developed the underlying method of computing the rotation of a vector in a Cartesian coordinate system and evaluating the length and angle of a vector. The CORDIC method was later expanded for multiplication, division, logarithm, exponential and hyperbolic functions.

IV. PIPELINED ARCHITECTURE

The principle of pipelining has emerged as a major architectural attribute of most present computer systems

.Pipelining is one form of imbedding parallelism or concurrency in a computer system. It refers to a segmentation of a computational process (say, an instruction) into several sub processes which are executed by dedicated autonomous units (facilities, pipelining segments) Parallel CORDIC can be pipelined by inserting registers between the adders stages. In most FPGA architectures there are already registers present in each logic cell, so pipeline registers has no hardware cost. Number of stages after which pipeline register is inserted can be modeled, considering clock frequency of system. When operating at greater clock period power consumption in later stages reduces due to lesser switching activity in each clock period.1

V. PROPOSED ALGORITHM FOR SCALING FREE CORDIC :

The proposed design is based on the following key ideas: 1) we use Taylor series expansion of sine and cosine functions to avoid scaling operation and 2) suggest a generalized sequence of micro-rotation to have adequate range of convergence (RoC) based on the chosen order of approximation of the Taylor series. A. Taylor Series Approximation of Sine and Cosine Functions The Taylor expansions of sine and cosine of an angle “-” are

given by

$$\sin a = (a) - (3!)^{-1} a^3 + (5!)^{-1} a^5 - \dots \cos a = 1 - (2!)^{-1} a^2 + (4!)^{-1} a^4 - \dots$$

We have estimated the maximum error in the evaluation of sine and cosine functions for different order of approximations. Therefore, we choose third order of approximation for Taylor’s expansion of sine and cosine functions.

1) Representation of Micro-Rotations Using Taylor Series Approximation:

Here, we study the impact of orders of approximation of Taylor series of sine and cosine functions on the micro- rotations to be used in CORDIC coordinate calculation. Both theoretical and simulation results are discussed to confirm the appropriate selection of the order of approximation. Using different orders of approximation of sin