Pre-computation of Rotation Bits in Unidirectional CORDIC for Trigonometric and Hyperbolic Computations

Satish Ravichandran, Vijayan Asari
Old Dominion University
Norfolk, Virginia 23529, USA
Email: sravi001@odu.edu, vasari@odu.edu

Abstract

A novel technique for the pre-computation of rotation bits for unidirectional CORDIC is proposed in this paper. The unidirectional CORDIC algorithms differs from the conventional CORDIC in the degree of rotation. A new technique is developed to pre-compute the rotation bits from any given angle. The hardware design VLSI circuit is implemented in a FPGA using Altera Quartus II VHDL. Experimental results obtained with computations of trigonometric and hyperbolic functions using the pre-computed bits show the accuracy results in the order of ~10^{-8}.

1. Introduction

CORDIC (COordinate Rotation Digital Computer) is an iterative algorithm to compute the values in the three systems, trigonometric, logarithmic and transcendental functions. The algorithm of CORDIC is derived from the two rotation equations [1],

$$x' = x\cos\phi - y\sin\phi \quad (1)$$

$$y' = y\cos\phi + x\sin\phi \quad (2)$$

The equations represent a vector (x, y) that is rotated by an angle ϕ. These equations are modified such that the rotations are characterized by an iterative shift operation in digital logic. Therefore, the modified equations are given by,

$$x_{i+1} = x_i - y_i .m.d_i.2^{-i} \quad (3)$$

$$y_{i+1} = y_i - x_i .m.d_i.2^{-i} \quad (4)$$

and

$$z_{i+1} = z_i - d_i.\tan^{-1}(2^{-i}) \text{ (for m =1)}; \quad (5)$$

$$z_{i+1} = z_i - d_i.\tanh^{-1}(2^{-i}) \text{ (m = -1)}; \quad (6)$$

where, m = 0, 1 or -1 which represents the rotation in the specific coordinate system (m=0 for linear, m = 1 for circular and m = -1 for hyperbolic co-ordinate system) and d_i = +1 or –1 which represents the sign of rotation (d_i = 1 for counterclockwise rotation and d_i = -1 for clockwise rotation)

From these equations it is seen that the rotation of any vector is represented in terms of simple shift and addition operation, thus making it less complex when architecture is built. From the CORDIC equations, by initializing x_o = gain and y_o = 0, the trigonometric and hyperbolic values after N iterations. In the conventional CORDIC the vector rotations will be in both directions based on the sign bit d_i. In the proposed unidirectional approach, the direction of rotation is either a '0' or a '1'. A '0' indicates no rotation of the vector and a '1', indicates a rotation of the vector.

2. Unidirectional CORDIC

The principle of unidirectional CORDIC is that any given angle can be represented using vector rotations only in the counter-clockwise direction. There are two important factors that affect the accuracy of the unidirectional CORDIC. It is understood from eqns (1) and (2) that eqns (3) and (4) are obtained only when the angles are restricted to $\tan(\phi) = 2^{-i}$ for trigonometric computations and $\tanh^{-1}(\phi) = 2^{-i}$ for hyperbolic computations. Therefore, there exists a difference in the given angle to the angle represented for the rotation. This difference is taken into consideration and the algorithm to compute the direction of rotation from a given angle.

Another important factor is gain, which is a constant for the conventional CORDIC But in the proposed unidirectional CORDIC algorithm, rotation occurs only at certain iterations (the occurrence of '1' in the counterclockwise direction of rotation). The gain factor is found each time the rotation bit is a '1', as per the equation $1/\sqrt{1+2^{-2i}}$ for trigonometric computation and as per the equation $1/\sqrt{1-2^{-2i}}$ for hyperbolic computation. If the value in rotation bit is a '0', then the multiplication is not required. The final value of this product yields the gain factor.

2.1. Unidirectional CORDIC algorithm

The steps involved in the computation of trigonometric and hyperbolic are the following.

Step 1: The angle for which the trigonometric values are to be computed is stored in a register and the

Proceedings of the IEEE Computer Society Annual Symposium on VLSI (ISVLSI'03)
0-7695-1904-0/03 $17.00 © 2003 **IEEE**

rotation bits are pre-computed for the given angle.

Step 2: The gain factor is selected from the ROM based on the direction bits.

Step 3: The initial values of register x_o and y_o are loaded with the initial gain factor and 0 respectively.

Step 4: The iteration is carried only for the direction bits '1'.

Step 5: The iterations are repeated until the final x_n and y_n are obtained.

3. Pre-computation of rotation bits

The basic idea behind unidirectional CORDIC algorithm is that the rotation is done only in the counterclockwise direction unlike the conventional CORDIC. Initially, the angle whose trigonometric values are to be computed is stored in a register. It can be seen that the first N/3 rotation bits are significant and so they are calculated separately [2]. The Least Significant Bits N/3+1 to N bits are found out from the remainder of the angle obtained after the computation of the first N/3 rotation bits. The corresponding value of the remainder is added to the remaining angle from the input angle bits (N/3+1 to N) to get the final rotation bit values.

The pre-computation of the rotation bits is different in the hyperbolic functions. This difference is because the value of the $\tanh^{-1}(x)$ of initial values is greater than x and so the computation was done for all the N bits. An algorithm is developed to pre-compute the rotation bits and is found that the second rotations are required for few angles. Subtracting the given angle with the constants stored compensated the difference of the angle, and constant value is too small that it has to be subtracted to reach the given angle. Architecture to compute the rotation bits in accordance to the proposed algorithm is shown in Fig. 1[2]. The architecture is designed and simulated using Altera Quartus 2 ver 1.1 VHDL. A special case of N=24 is selected to simulate the architecture. Table 1, shows the partial result of the comparison of trigonometric values for conventional and unidirectional methods. The error between the two results is in the order of 10^{-8}.

4. Conclusion

A novel, modified CORDIC algorithm is obtained by performing unidirectional vector rotations. Experiments performed on calculating the trigonometric values and hyperbolic values show that the degree of accuracy has been maintained. The proposed technique has the advantage of higher speed as the number of iterations has been decreased with least circuit complexity and can be applied to various digital signal processing or neural network applications, which require intense mathematical computation.

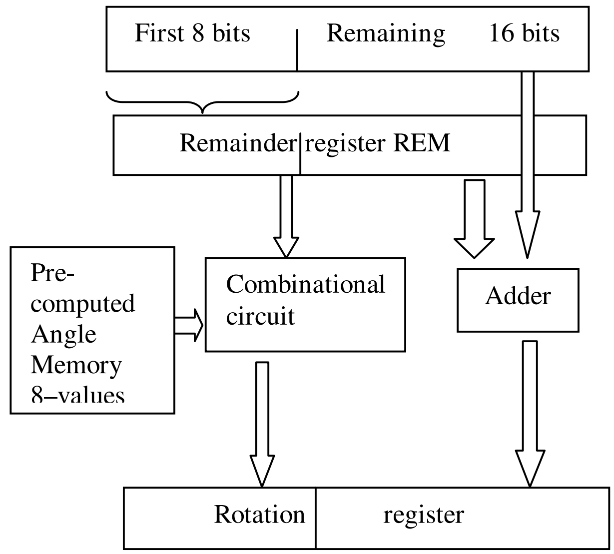

Figure 1. Pre-computation signed digits evaluation architecture.

	Angle	Value	Error for cosine	Error for sin
cos	1	0.999848	-3.15E-07	1.06E-07
cos	5.5	0.995396	1.82E-07	5.6E-08
cos	13	0.97437	5.4E-08	7.8E-08
cos	28	0.882948	-4.37E-07	-3.96E-07
cos	33.5	0.833886	-1.59E-07	-5.5E-08

Table 1. Comparison of results with conventional CORDIC

References

[1] Ray Andraka, "A survey of CORDIC algorithms for FPGAs," FPGA '98, Proceedings of the 1998 ACM/SIGDA sixth international symposium on Field programmable gate arrays, Monterey, CA, Feb. 1998, pp191-200.

[2] Satish Ravichandran and Vijayan K. Asari, "VLSI Architecture for Pre-computation of Rotation Bits in Unidirectional Flat-CORDIC," *Proceedings of 10th NASA Symposium on VLSI Design*, Albuquerque, New Mexico, USA, Mar 2002, pp. 3.2.1-3.2.6