International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 11 Issue 3 –NOVEMBER 2014.

Fixed Angle of Rotation Using CORDIC Designs

*1K.Vinod Kumar and #2K.Venkanna

*Student, Dept of ECE (VLSI), Sree Vahini Institute of Science and Technology, Tiruvuru., A.P, India

#Asst Professor, Dept of ECE (VLSI), Sree Vahini Institute of Science and Technology, Tiruvuru., A.P, India

1kvinod416@gmail.com

2venkanna.kota3@gmail.com

Abstract— Rotation of vectors through fixed and known angles has wide applications in robotics, digital signal processing, graphics, games, and animation. But, we do not find any optimized coordinate rotation digital computer (CORDIC) design for vector-rotation through specific angles. Therefore, in this paper, we present optimization schemes and CORDIC circuits for fixed and known rotations with different levels of accuracy. For reducing the area- and time-complexities, we have proposed a hardwired pre-shifting scheme in barrel-shifters of the proposed circuits. Two dedicated CORDIC cells are proposed for the fixed-angle rotations. In one of those cells, micro-rotations and scaling are interleaved, and in the other they are implemented in two separate stages. Pipelined schemes are suggested further for cascading dedicated single-rotation CORDIC units for high-throughput and reduced latency implementations. We have the optimized set of micro-rotations for fixed and known angles. The derived optimized scale-factors from a set of micro rotations and dedicated shift-add circuits are used to implement the scaling. We have synthesized the proposed CORDIC cells using Xilinx field programmable gate array platform and shown that the proposed design offer higher throughput and less area-delay product than the reference CORDIC design for fixed and known angles of rotation.

Index Terms- Coordinate rotation digital computer (CORDIC), numerically controlled oscillator (NCO), look up table (LUT)

I. INTRODUCTION

The name CORDIC is nothing but iterative based mathematics for coordinated rotation of digital computer. In 1959, Volder was the scientist who developed CORDIC iterative form of computation for trigonometry functions, multiplication and division. Over the period of time many application of CORDIC has been formed with variety of algorithm for high performance and low cost hardware solutions. This paper stresses on design of CORDIC in terms fixed angle rotation. For known and fixed angel of vector rotation has wide applications in graphics, game, robotics and robotics. Modulation and demodulation can be performed by successive vector rotation through fixed angels and translation of the links. The translation operation is realized by simple addition of old coordinated values to obtain new coordinates for rotational steps to be accomplished by suitable successive rotation of CORDIC circuit for fixed rotation through know small angels. There are plenty of examples of uniform rotation starting from electrons inside an atom to the planets and satellites. A simple example of uniform rotations is the hands of an animated mechanical clock which perform one degree rotation each time. There are several cases where high-speed constant rotation is required in games, graphic, and animation. In modelling, simulation, animation and games are area where objects with constant rotation are more often used. Dedicated CORDIC circuits are used to implement efficiently through known small angle rotation. In some areas like signal processing, engineering application, communication by multiplication of complex number with a known complex constants. Previously, CORDIC circuits were developed for implanting complex multiplication which is more often used in digital signal processing (DSP) application[16]-[18], but there is no detailed information with respect to efficiency of CORDIC realization of fixed and known angle rotations and constant complex multiplication. Latency of computation is the major issue with the implementation of CORDIC algorithm due to its linear-rate convergence [19]. It requires (n+1) iterations to have -bit precision of the output. Overall latency of computation increases linearly with the product of the word-length and the CORDIC iteration period. The speed of CORDIC operations is, therefore, constrained either by the precision requirement (iteration count) or the duration of the clock period. The angle recoding (AR) schemes [5]–[9] could be applied for reducing the iteration count for CORDIC implementation of constant complex multiplications by encoding the angle of rotation as a linear combination of a set of selected elementary angles of micro rotations.

However, in AR methods, this constraint is relaxed by adding zero into the linear combination to obtain the desired angle. AR method is based on the greedy algorithm that tries to represent the remaining angle using the closest elementary angle. Using these recoding schemes the total number of iterations could be reduced to less than half of the conventional CORDIC algorithm for the same accuracy. Wu et al. [7] have suggested an AR scheme based on an extended elementary-angleset (EEAS) that provides a more flexible way of decomposing the target rotation angle. .EEAS has better recoding efficiency in terms of the number of iterations and can yield better error performance than the AR scheme based on EAS. But the iteration period for EEAS is longer, and involves double the numbers of adders/subtracters in the CORDIC cell compared with that of the other. Most of the advantages gained in the AR schemes are cancelled out by the hardware and time involved in scaling the pseudo-rotated vector.

It is desirable to have exhaustive search than greedy search for elementary angle set (EAS) due to the fact that angle of rotation for fixed rotation case is known a priori. Moreover, the barrel shifter complexity is reduced to almost half of that of a CORDIC. So thus we have proposed some techniques to minimize complexity of barrel shifter. As CORDIC is a sequential process it is desirable to use pipelined implementation which is yet to be exploited but this makes it out of parallel process execution. This paper tries to present optimization schemes for reducing the number of iterations of micro rotation and even reducing the complexity of barrel shifter for vector rotation in case of fixed angle. We also aim for cascaded pipelined circuit for future use, which is expected to be involving less area delay complexity and faster than the existing approaches.

The contribution of paper are listed as below

1. Implementation of fixed angle vector rotation with optimized set of micro-rotation.
2. Scaling circuits are derived from shift-add operation.
3. To reduce barrel shifter complexity, a hardware pre-shifting scheme is been introduced.
4. Single rotation CORDIC circuits are designed.

The rotation-mode CORDIC algorithm to rotate a vector U=[UXUY]T through an angle Φ to obtain a rotated vector V=[VXVY]T is given by [1], [2]

Such that when n is sufficiently large Where if σi = -1 then Φi < 0 and σi=1 otherwise, and is the scale-factor of the CORDIC algorithm, given by

For fixed angle rotation case, the pre-computed value Φi and σi indicating sign bit can be stored in sign bit register (SBR) in CORDIC circuit. This during the CORDIC iteration, CORDIC circuit need not compute rest of angle Φi.[3].

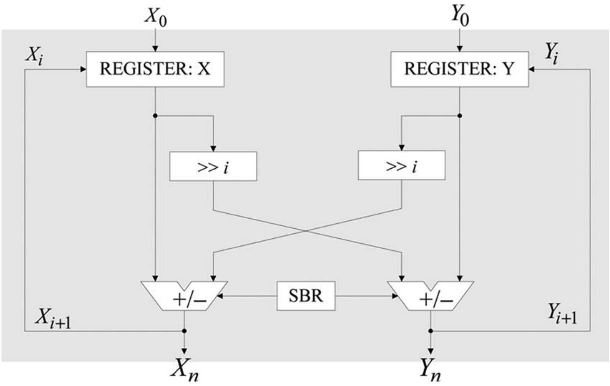

Fig. 1. Reference CORDIC circuit for fixed rotations

As shown in fig.1. and with respect to (1a) and (1b) a reference CORDIC circuit for fixed rotation X0 and Y0 can b fed as set or reset input in to the input register and hence later feedback values Xi and Yi are even fed parallel to input register at ith iteration. So with the conventional concept of CORDIC we feed in initial values X0 and Y0 as well as feedback values Xi and Yi via a pair of MUX.

Here we can find a set of small number of predetermined elementary angle {αi, for 0 ≤ i ≤ m-1}, where αi = arctan (2-k(i)) for known and fixed angle rotation of vector through rotation mode CORDIC circuit which are used in ith micro rotation of CORDIC algorithm(1), where m is minimum number of micro-rotation. The rotation through any angle in 0 ≤ θ ≤ 2π, can be mapped on to positive rotation 0 ≤ Φ≤ π/4 without much arithmetic operation [10]. Hence to perform optimization rotation mapping is done so that rotation angle lies in the range of 0 ≤ Φ ≤ π/4.

The simple pseudo code to optimize a set of micro-rotations is described in Algorithm 1. If the maximum accuracy which is defined as the maximum tolerable error between desired angle and approximated angle is given as an input, the optimization algorithm searches the parameters and that can minimize an objective function. The algorithm starts with the single micro-rotation, i.e., , then if the micro-rotation that has smaller angle of deviation than cannot be found, the number of micro-rotations is increased by one and the optimization algorithm is run again. Exhaustive search is employed in the optimization algorithm to search the entire parameter space for all the combinations of and . Based on the obtained micro-rotations, the parameters for scaling operation can be searched with the different objective function, which is described in Section IV. The sub-optimal set of micro-rotations may be used in some cases, if the optimal set of micro-rotations cannot satisfy the design constraint for scaling. We have used sub-optimal solutions particularly for the rotation with the angle of 31 and 35 in Table I since the scaling requires more terms in these two cases if optimal solutions are used.

ϕ°	$k(0), s_0$	$k(1), s_1$	$k(2), s_2$	$k(3), s_3$	$\Delta\phi$
45	0, 1	−−	−−	−−	0.000
43	0, 1	5, 0	8, 0	−−	0.014
41	0, 1	4, 0	7, 0	−−	0.024
39	0, 1	3, 0	6, 1	8, 1	0.006
37	0, 1	3, 0	6, 0	−−	0.020
35	2, 1	2, 1	2, 1	3, 0	0.016
33	1, 1	3, 1	7, 0	8, 0	0.019
31	1, 1	4, 1	6, 1	−−	0.037
29	2, 1	2, 1	6, 1	−−	0.032
27	1, 1	7, 1	−−	−−	0.013
25	1, 1	5, 0	8, 1	−−	0.001
23	1, 1	4, 0	−−	−−	0.011
21	2, 1	2, 1	3, 0	10, 1	0.003
19	1, 1	3, 0	7, 0	−−	0.008
17	1, 1	2, 0	4, 1	6, 1	0.000
15	2, 1	6, 1	10, 1	−−	0.013
13	1, 1	2, 0	7, 1	−−	0.024
11	3, 1	4, 1	8, 1	10, 1	0.019
9	3, 1	5, 1	9, 1	−−	0.027
7	3, 1	9, 0	−−	−−	0.013
5	3, 1	5, 0	7, 0	9, 1	0.001
3	4, 1	7, 0	9, 0	−−	0.017
1	6, 1	9, 1	−−	−−	0.007

s_i is the sign-bit corresponding to the sign term σ_i, such that $s_i = 1$ and 0 for $\sigma_i = 1$ and -1, respectively. $\Delta\phi = |\phi - \phi_A|$.

TABLE I : OPTIMIZATION OF FULL ROTATIONS WITH FOUR MICRO-ROTATIONS

ϕ°	$k(0), s_0$	$k(1), s_1$	$k(2), s_2$	$k(3), s_3$	$\Delta\phi$
2.0	5, 1	8, 1	12, 0	−−	0.0003
1.9	5, 1	9, 1	−−	−−	0.0018
1.8	5, 1	13, 1	−−	−−	0.0031
1.7	5, 1	9, 0	12, 1	13, 1	0.0010
1.6	5, 1	8, 0	11, 1	13, 1	0.0011
1.5	1, 0	2, 1	2, 1	13, 0	0.0004
1.4	6, 1	7, 1	10, 1	−−	0.0013
1.3	5, 1	7, 0	11, 0	12, 0	0.0003
1.2	5, 1	7, 0	9, 0	11, 0	0.0024
1.1	6, 1	8, 1	12, 0	14, 0	0.0015
1.0	6, 1	9, 1	13, 0	−−	0.0001
0.9	6, 1	14, 1	−−	−−	0.0013
0.8	6, 1	9, 0	12, 1	−−	0.0027
0.7	7, 1	8, 1	11, 1	−−	0.0006
0.6	6, 1	8, 1	10, 0	12, 0	0.0014
0.5	7, 1	10, 1	−−	−−	0.0036
0.4	7, 1	10, 0	13, 1	−−	0.0013
0.3	7, 1	9, 0	11, 0	13, 0	0.0007
0.2	8, 1	11, 0	14, 1	−−	0.0007
0.1	9, 1	12, 0	−−	−−	0.0021

TABLE II : OPTIMIZATION OF SMALL ROTATIONS WITH FOUR MICRO-ROTATIONS

In Table II, it is shown further that rotations through in an interval of 0.1 could be obtained by four micro-rotations with angular deviation, . Here we can make an observation that we can always achieve higher accuracy with more number of micro-rotations. From Table II, we find that higher accuracy could be achieved in case of small rotation angles like 1 or 2 , compared to the most of the larger angles when the same number of micro-rotations is used.

II. IMPLEMENTATION OF MICRO-ROTATIONS

Since the elementary angles and direction of micro-rotations are predetermined for the given angle of rotation, the angle estimation data-path is not required in the CORDIC circuit for fixed and known rotations. Moreover, because only a few elementary angles are involved in this case, the corresponding control-bits could be stored in a ROM of few words. A CORDIC circuit for complex constant multiplications is shown in Fig. 2. The ROM contains the control-bits for the number of shifts corresponding the micro-rotations to be implemented by the barrel-shifter and the directions of micro-rotations are stored in the sign-bit register (SBR). The major contributors to the hardware-complexity in the implementation of a CORDIC circuit are the barrel-shifters and the adders. There are several options for the implementation of adders [22], from which a designer can always choose depending on the constraints and requirements of the application. But, we have some scope to develop techniques for reducing the complexity of barrel-shifters over the conventional designs as discussed in the followings.

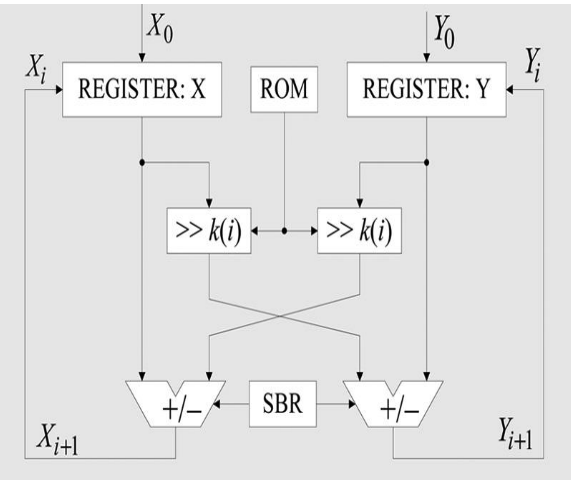

Fig. 2. CORDIC cell for constant complex multiplications

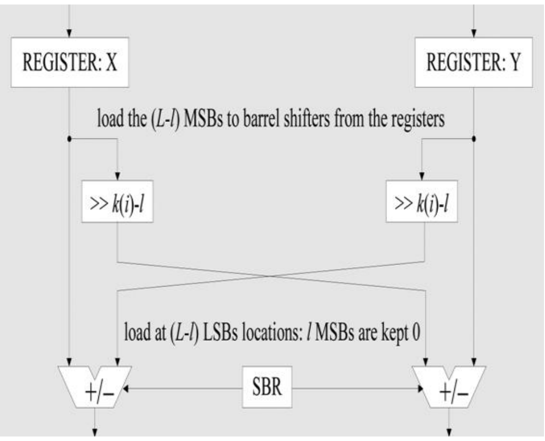

Fig. 3. Hardwired pre-shifting in basic CORDIC module

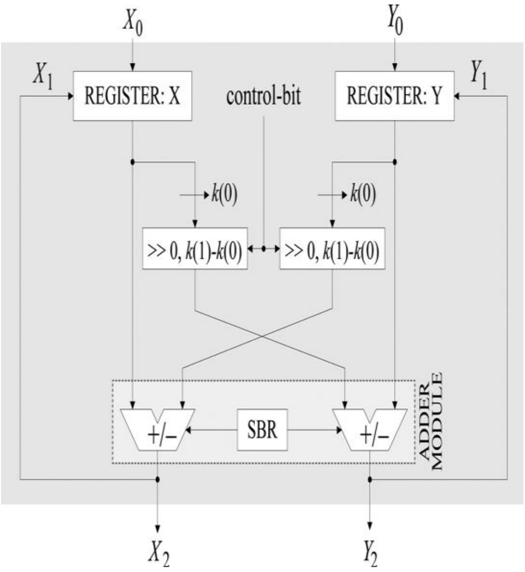

Fig. 4. Hardwired pre-shifted bi-rotation CORDIC circuit

For reduction of adder complexity over the cascaded single-rotation CORDIC, the micro-rotations could be implemented by a cascaded bi-rotation CORDIC circuit. A two-stage cascaded bi-rotation CORDIC is shown in the above figure The first two of the micro-rotations as shown in Table I out of the four-optimized micro-rotations could be implemented by stage-1, while the rest two are performed by stage-2. The structure and function of the bi-rotation CORDIC is shown in Fig. 4. For implementing six selected micro-rotations, we can use a three-stage-cascade of bi-rotation CORDIC cells. The three-stage bi-rotation cells could however be extended further when higher accuracy is required.

III. CONCLUSION

The number of micro-rotations for rotation of vectors through known and fixed angles is optimized and several possible dedicated circuits are explored for rotation-mode CORDIC processing with different levels of accuracy. The proposed CORDIC cell with interleaved scaling involves more area, but offers more throughputs and involves nearly less latency and less ADP, than the reference design for known and fixed rotations. The proposed single-rotation cascade and irritation cascade require, respectively, and times more area over the reference design, but offer nearly 16.3 and 7.0 times more throughput, and involve nearly 4.6 and 2.5 times less ADP with nearly half and two-thirds of the latency of the other. With progressing scaling trends, since the silicon area is getting continually cheaper, it appears to be a good idea to use the cascaded designs for their potential for high-throughput and low-latency implementation. It is found that higher accuracy could be achieved in case of smaller angles of rotation when the same number of micro-rotations is used. The small angle rotators could therefore be very much useful for shape design and curve tracing for animation and gaming devices. The fixed-angle CORDIC rotation would have wide applications in signal processing, games, animation, graphics and robotics, as well.

IV. REFERENCES

[1] J. E. Volder, “The CORDIC trigonometric computing technique,” IRE Trans. Electron. Comput., vol. EC-8, pp. 330–334, Sep. 1959.
[2] J. S. Walther, “A unified algorithm for elementary functions,” in Proc.38th Spring Joint Comput. Conf., 1971, pp. 379–385.
[3] Y. H. Hu, “CORDIC-based VLSI architectures for digital signal processing,” IEEE Signal Process. Mag., vol. 9, no. 3, pp. 16–35, Jul. 1992.
[4] P. K. Meher, J. Valls, T.-B. Juang,K. Sridharan, andK. 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.
[5] Y. H. Hu and S. Naganathan, “An angle recoding method for CORDIC algorithm implementation,” IEEE Trans. Comput., vol. 42, no. 1, pp. 99–102, Jan. 1993.
[6] Y. H. Hu and H. H.M. Chern, “A novel implementation of CORDIC algorithm using backward angle recoding (BAR),” IEEE Trans.Comput., vol. 45, no. 12, pp. 1370–1378, Dec. 1996.
[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.
[8] T.-B. Juang, S.-F. Hsiao, and M.-Y. Tsai, “Para-CORDIC: Parallel cordic rotation algorithm,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 51, no. 8, pp. 1515–1524, Aug. 2004.
[9] T. K. Rodrigues and E. E. Swartzlander, “Adaptive CORDIC: Using parallel angle recoding to accelerate CORDIC rotations,” in Proc. 40th Asilomar Conf. Signals, Syst. Comput. (ACSSC), 2006, pp. 323–327.
[10] 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. for Video Technol., vol. 15, no. 11, pp. 1463–1474, Nov. 2005.

[11] C. Y. Kang and E. E. Swartzlander, Jr., “Digit-pipelined direct digital frequency synthesis based on differential CORDIC,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 53, no. 5, pp. 1035–1044, May 2006.